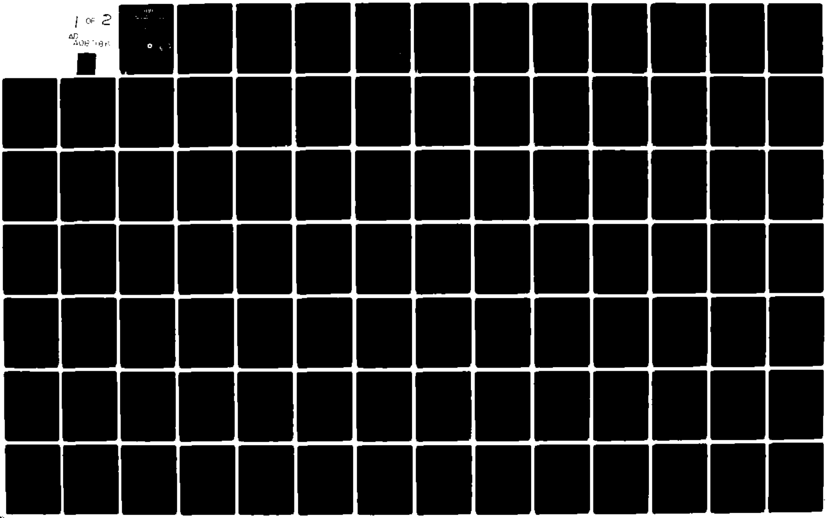
AD-A087 186 ARMY ENGINEER WATERWAYS EXPERIMENT STATION VICKSBURG--ETC F/G 1/5
PAVEMENT EVALUATION AND OVERLAY DESIGN USING VIBRATORY NONDESTR--ETC(U)
MAR 80 R A WEISS DOT-FA73WAI-377

UNCLASSIFIED FAA-RD-77-186-VOL-1 NL

1 OF 2
AD A087186



Report No. FAA-RD-77-186-I

LEVEL II

PAVEMENT EVALUATION AND OVERLAY DESIGN USING VIBRATORY NONDESTRUCTIVE TESTING AND LAYERED ELASTIC THEORY

Volume I

Development of Procedure

Richard A. Weiss

U. S. Army Engineer Waterways Experiment Station
Geotechnical Laboratory
P. O. Box 631, Vicksburg, Miss. 39180

DTIC
ELECTE
JUL 25 1980
S C D

MARCH 1980
FINAL REPORT

Document is available to the public through the
National Technical Information Service,
Springfield, Va. 22151

Prepared for

U. S. DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION
Systems Research & Development Service
Washington, D. C. 20591

ADA087186

DDC FILE COPY

80 7 24 015

NOTICES

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the object of this report.

19 77-186-VOL-1

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
18 FAA-RD-77-186-I	AD-A087 186	

4. Title and Subtitle

6 PAVEMENT EVALUATION AND OVERLAY DESIGN USING VIBRATORY NONDESTRUCTIVE TESTING AND LAYERED ELASTIC THEORY. Volume I. DEVELOPMENT OF PROCEDURE.

5. Report Date
March 1980 411412

6. Performing Organization Code

7. Author(s)

10 Richard A. Weiss

8. Performing Organization Report No.
11 Mar 80

9. Performing Organization Name and Address

U. S. Army Engineer Waterways Experiment Station
Geotechnical Laboratory
P. O. Box 631, Vicksburg, Miss. 39180

10. Work Unit No. (TRAIS)

11. Contract or Grant No.
15 DOT-FA73WAI-377

12. Sponsoring Agency Name and Address
12 176

U. S. Department of Transportation
Federal Aviation Administration
Systems Research and Development Service
Washington, D. C. 20591

13. Type of Report and Period Covered
9 Final Report
Oct[redacted]77 [redacted]Dec[redacted]978

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A procedure is developed for determining the allowable load-carrying capacities and the required overlay thicknesses of airport pavements. A layered elastic theory approach is used with vibratory nondestructive tests supplying the dynamic responses of pavements. For a given pavement, a computer program SUBE is used to determine the value of the subgrade Young's modulus from the measured dynamic responses, and a computer program PAVEVAL, which is based on the layered elastic theory, is used to calculate the allowable load-carrying capacity and the required overlay thickness. Limiting subgrade strains and horizontal stresses in pavement layers are used as criteria for determining load-carrying capacities and overlay thickness requirements. Single- and multiple-wheel loadings are considered. Volume II of this report presents a validation of these procedures for three airport sites.

17. Key Words	18. Distribution Statement
Nondestructive testing Layered elastic theory Pavement design and evaluation	Document is available to the public through the National Technical Information Service, Springfield, Va. 22151

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	174	

411412

METRIC CONVERSION FACTORS

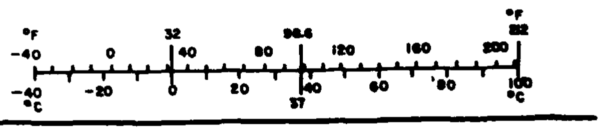
Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
	LENGTH			
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
	AREA			
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
	MASS (weight)			
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
	VOLUME			
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
	TEMPERATURE (exact)			
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
	LENGTH			
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
	AREA			
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
	MASS (weight)			
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
	VOLUME			
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
	TEMPERATURE (exact)			
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



PREFACE

This study was conducted during the period October 1977 to December 1978 by personnel of the Geotechnical Laboratory (GL), U. S. Army Engineer Waterways Experiment Station (WES), for the U. S. Department of Transportation, Federal Aviation Administration, as a part of Inter-Agency Agreement No. DOT FA73WAI-377, "New Pavement Design Methodology."

The study was conducted under the general supervision of Messrs. J. P. Sale and R. G. Ahlvin, Chief and Assistant Chief, respectively, of GL; R. L. Hutchinson and H. H. Ulery, Jr., Chief and Principal Technical Advisor, respectively, of the Pavement Systems Division; and under the direct supervision of A. H. Joseph, Chief of the Engineering Investigation Testing and Validation Group; and J. W. Hall, Jr., Chief of the Prototype Testing and Evaluation Unit. The programming for this study was accomplished in part by Mr. Ricky Austin, Research and Analysis Group. Significant contributions were made by Messrs. J. L. Green and A. J. Bush III of the Prototype Testing and Evaluation Unit, and by Dr. W. R. Barker of the Research and Analysis Group. The report was written by Dr. R. A. Weiss.

COL John L. Cannon, CE, and COL Nelson P. Conover, CE, were Directors of the WES during the conduct of this study and the preparation of this report. The Technical Director was Mr. F. R. Brown.

Accession For
NTIS GRA&I
DDC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or special
A

TABLE OF CONTENTS

PRECEDING PAGE BLANK-NOT FILMED

INTRODUCTION

BACKGROUND

The increasing expense of pavement construction and rehabilitation makes it essential to have a fast and reliable method of accurately predicting the allowable load-carrying capacity and the required overlay thickness for pavement upgrading. Vibratory nondestructive testing can play an important part for the rapid evaluation of airport pavements.[1-6] The U. S. Army Engineer Waterways Experiment Station (WES) was requested to develop a pavement evaluation method based on vibratory nondestructive testing combined with layered elastic theory. This study combines the layered elastic theory for calculating stress and strain in a pavement with the nonlinear theory of dynamic pavement response that describes the vibratory nondestructive test data to produce a working method, including computer programs, for evaluating pavements and designing overlays.

The combined method of layered elasticity theory and vibratory nondestructive testing is compared with the conventional method that uses the California Bearing Ratio (CBR) for evaluating asphaltic concrete (AC) pavements and with the Westergaard method for evaluating portland cement concrete (PCC) pavements.[7] It is also compared with the pavement evaluation method that uses a correlation between the strength of a pavement and the dynamic stiffness modulus (DSM) that is obtained from vibratory nondestructive testing.[1]

The CBR and Westergaard methods require destructive tests to measure the CBR and the coefficient of subgrade reaction, respectively. To circumvent the destructive tests, a vibratory nondestructive test method, which directly correlates the allowable load-carrying capacity and the required overlay thickness to a mechanical impedance that is measured at the pavement surface (the DSM), was developed at the WES for evaluating AC and PCC pavements.

The DSM is obtained from vibratory nondestructive test data that are obtained with the WES electrohydraulic vibrator, which can

generate dynamic loads up to 15 kips with a constant 16-kip static load (WES 16-kip vibrator) and a constant frequency of 15 Hz.[4] These data consist of dynamic load-deflection curves that are measured at the pavement surface. The dynamic load-deflection curves are nonlinear in general, and the DSM is the slope of the dynamic load-deflection curve for a dynamic load of about 10-14 kips. The measured DSM is corrected to a common pavement temperature of 70°F, and the corrected value of the DSM is correlated to the allowable load-carrying capacity and the required overlay thickness of a pavement.[1,6] The DSM method is empirical and does not take into consideration the layered elastic structure of the pavement or the interface conditions between the pavement layers.

In order to improve on the method of directly correlating pavement performance with vibratory nondestructive test data, an attempt was made to combine the layered elastic theory of pavements with the pavement impedance values measured by vibratory nondestructive tests. In this way, the pavement structure could be considered. The layered elastic model of pavements required the Young's modulus and the Poisson's ratio of the subgrade and pavement layers to be known. The elastic moduli of the pavement layers are estimated by various means, and only the subgrade Young's modulus is obtained by vibratory nondestructive tests.

The pavement evaluation method presented herein consists of determining the subgrade Young's modulus from the dynamic response of a pavement measured by vibratory nondestructive tests and using the determined value of the subgrade Young's modulus in the layered elastic theory to calculate the allowable load-carrying capacity and the required overlay thickness of a pavement. Two computer programs, SUBE and PAVEVAL, are used for the necessary computations and to obtain the results.

The subgrade Young's modulus is determined from dynamic load-deflection curves that are measured at the pavement surface. In general, these dynamic load-deflection curves are nonlinear, and a nonlinear dynamic theory is required to extract the value of the subgrade Young's modulus from these measured curves. The nonlinear

dynamic theory is used to remove the extraneous effects of the static and dynamic loads developed by the vibrator on the predicted values of the subgrade Young's modulus.[3,4] The value of the subgrade Young's modulus used for calculating the allowable load-carrying capacity and the required overlay thickness of a pavement should reflect only the stress conditions in the sugrade due to the aircraft loading and the natural overburden pressure. The computer program SUBE was developed from the nonlinear theory of pavement response to dynamic loads and is used to determine the subgrade Young's modulus from the measured dynamic load-deflection curves.

Within the context of the layered elastic theory, pavements are represented by a stack of elastic layers, the subgrade being of infinite extent. This layered elastic theoretical model of a pavement structure is used to calculate the elastic stress and strain at any point in the pavement or the subgrade. Each pavement layer is characterized by a Poisson's ratio (ν), a Young's modulus (E), and a layer thickness (h). The Shell BISAR computer program is based on the layered elastic theory and relates the stress and the strain in each pavement layer to the static load applied to the surface of a pavement. The computer program PAVEVAL that is used for pavement evaluation and overlay design is a modification of the BISAR program. Figure 1 represents a typical pavement structure subjected to a loading according to the layered elastic theory approach.

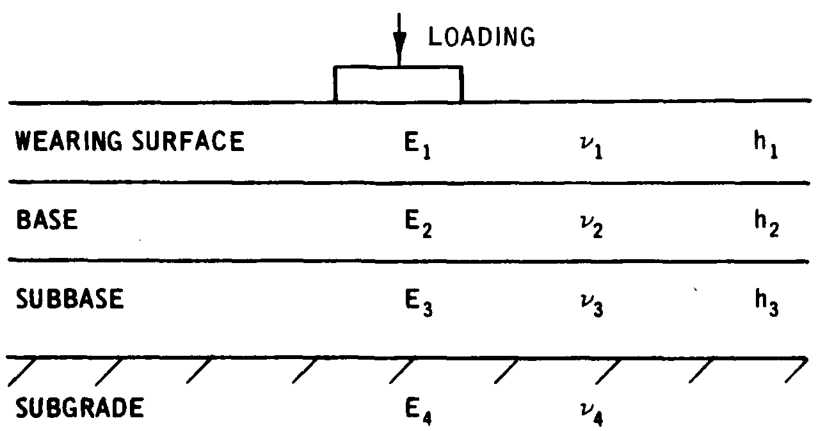


Figure 1. Typical pavement structure with loading according to the layered elastic theory

The computer programs SUBE and PAVEVAL were developed on the IBM 360/65 computer and are designed for practical use by pavement engineers.

OBJECTIVES

A pavement evaluation procedure is required that will use vibratory nondestructive testing and analysis of data to obtain the value of the subgrade modulus for input into the layered elastic theory for calculating stresses and strains in a pavement. The basic objectives of this study are:

a. To further develop and evaluate a theoretical procedure for determining the subgrade Young's modulus from vibratory nondestructive test data.

b. To determine the allowable load-carrying capacity for AC and PCC pavements for single- and multiple-wheel loadings using the subgrade modulus in the layered elastic theory.

c. To determine the overlay thickness required to upgrade AC and PCC pavements for single- and multiple-wheel loadings using elastic pavement parameters calculated from vibratory nondestructive testing results in the layered elastic theory.

SCOPE

To achieve these objectives, theoretical and experimental work was done.

THEORETICAL STUDIES

The theoretical studies included:

a. A logical method of selecting the values of the elastic moduli of each pavement layer.

b. The development of the nonlinear dynamic computer program SUBE to predict the values of the subgrade Young's modulus from measured vibratory nondestructive test data.

c. The determination of the limiting vertical strain in the subgrade and the limiting tensile strain in the AC layer of AC pavements, and a limiting tensile stress criterion in PCC pavements as design criteria to be used with the layered elastic pavement model.

d. The development of the PAVEVAL layered elastic computer program to calculate the allowable load-carrying capacity and the required overlay thickness for AC and PCC pavements

with single- and multiple-wheel loadings. A comparison with conventional CBR and Westergaard methods is made.

EXPERIMENTAL STUDIES

Dynamic load-deflection curves and CBR values were measured in the field for PCC and AC pavements.

MATERIAL PARAMETERS AND FAILURE CRITERIA REQUIRED FOR PAVEMENT EVALUATION AND OVERLAY DESIGN

GENERAL CONSIDERATIONS

Repeated aircraft loadings on a pavement will eventually lead to a failure of the pavement. The ultimate purpose of the nondestructive testing of a pavement is to estimate the allowable load-carrying capacity of a pavement for a specified number of yearly load repetitions or to determine the overlay thickness required to upgrade a pavement when the operating aircraft weight and yearly number of load repetitions are specified.[1] The estimation of the allowable load-carrying capacity and the required overlay thickness requires a knowledge of the failure processes that occur in AC and PCC pavements. Vibratory nondestructive testing should supply some of the pavement parameters that enter into the physical description of the failure processes.[2-4]

Pavements fail for a variety of reasons. Many pavements fail because the pavement does not properly protect the subgrade from large stresses and strains that can cause excessive plastic and elastic deformation of the soil in the subgrade. Experience has shown that the condition of failure in AC pavements may be described by a limiting elastic (resilient) vertical strain in the top of the subgrade and a limiting tensile strain at the bottom of the AC pavement layer, while the condition of failure in PCC pavements can be described by a limiting tensile stress at the bottom of the PCC layer.[8,9] These limiting values of stress and strain are related to the allowable load-carrying capacity and the required overlay thickness of a pavement through the structure of the pavement, i.e., through the thickness and the material type of each layer of the pavement and the subgrade.

The materials in the pavement layers must be described by material parameters, which determine the stress-strain characteristics. The proper mechanical parameters chosen to describe the pavement material will depend on the type of problem under consideration. For instance, if the time history of the plastic flow of the pavement

material is of interest, then some plastic flow parameters relating permanent strain to the operating stress must be introduced. If the resilient properties or the incipient plastic flow characteristics of pavement materials are of interest, the Young's modulus and the Poisson's ratio of the subgrade and pavement layers are sufficient for a complete description.

FAILURE IN AC PAVEMENTS

The failure of AC pavements generally occurs by two processes: (a) cracking of the bituminous wearing surface and (b) rutting of the wearing surface along the wheel paths.[9] The fatigue cracking along the wearing surface due to repeated flexural loadings is determined by the magnitude of the tensile strain at the bottom of the wearing surface, while the rutting of the wearing surface may be governed in part by the vertical compressive strain at the top of the subgrade and by the flow of the AC material. Therefore, in this study, the allowable load-carrying capacity and the required overlay thickness for AC pavements will be determined mainly by a limiting vertical compressive strain at the top of the subgrade. However, this may lead to erroneous values because of the gross oversimplifications involved.

For a given load at the pavement surface, the values of the stress and the strain in the pavement and the subgrade depend on the Young's modulus and the Poisson's ratio of the subgrade and each pavement layer. Therefore, if the elastic moduli of the pavement layers are known, it is the Young's modulus of the subgrade that is the unknown parameter determining stress and strain in the pavement and the subgrade. This parameter must be obtained by vibratory nondestructive testing.

FAILURE IN PCC PAVEMENTS

It is assumed herein that PCC pavements fail because of fatigue cracking associated with the repeated flexural stress in the PCC layer. Actually, many failures occur at joints, but this condition is not considered in this study. The fatigue cracking of the wearing surface of PCC pavements is governed by the tensile stress at the bottom of the

wearing surface, and the value of this stress, for a given operating load at the pavement surface, is determined by the elastic moduli of the subgrade and pavement layers.[9] Assuming that the elastic moduli of the pavement layers are known, it is the subgrade Young's modulus that is the unknown parameter determining the operating value of the tensile stress at the bottom of the PCC layer. This elastic parameter must be supplied by vibratory nondestructive testing.

PAVEMENT EVALUATION AND OVERLAY DESIGN

The computer program PAVEVAL was written to incorporate the material parameters and the limiting stress and strain criteria into a procedure for calculating the allowable load-carrying capacity and the overlay thickness required for pavement upgrading. PAVEVAL, used in conjunction with the computer program SUBE that predicts the value of the subgrade Young's modulus, was developed to be a practical tool for the pavement engineer to use for evaluation and overlay design purposes. Detailed descriptions and listings of the computer programs SUBE and PAVEVAL are given in Appendixes A and B, respectively. Figure 2 gives a flow diagram of the general procedure used for pavement evaluation and overlay design.

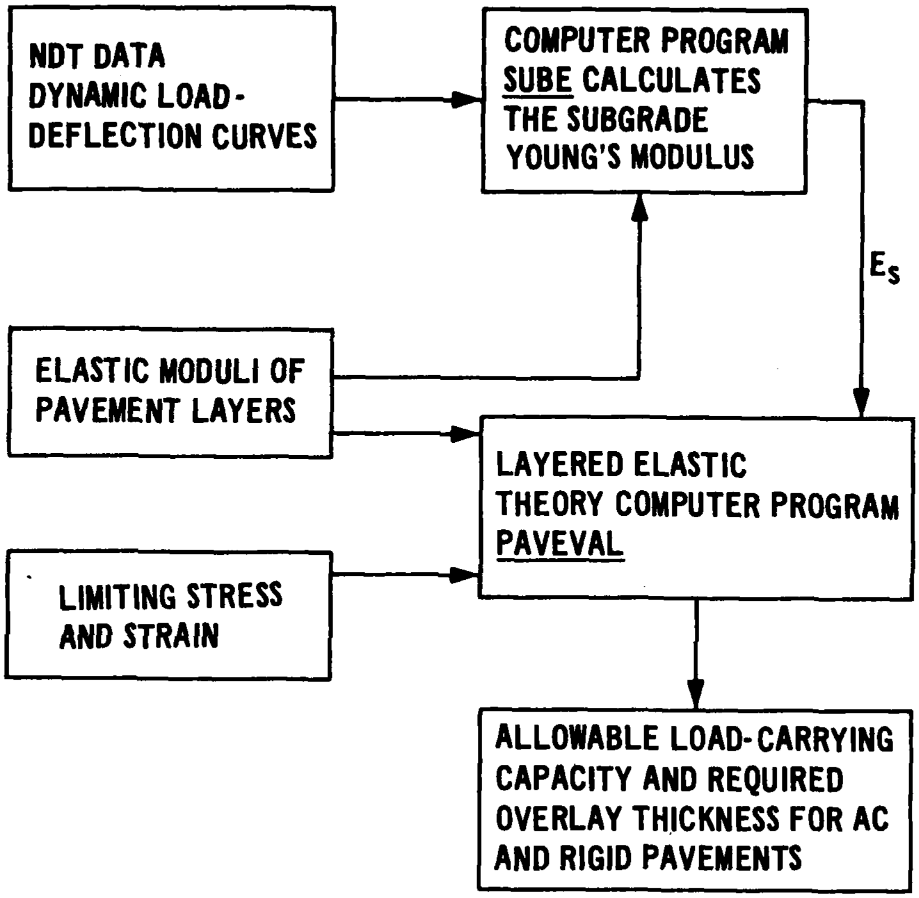


Figure 2. Pavement evaluation and overlay design by the combined methods of layered elastic theory and vibratory nondestructive testing

SELECTION OF THE PAVEMENT AND AIRCRAFT PARAMETERS FOR THE COMPUTER PROGRAMS SUBE AND PAVEVAL

GENERAL CONSIDERATIONS

The pavement parameters required by the computer programs SUBE and PAVEVAL are the Young's modulus, the Poisson's ratio, the thickness of the pavement layers and the subgrade, and the flexural strength for the PCC layers. Some progress has been made toward determining all of the elastic moduli of the pavement layers by vibratory nondestructive testing, but the results are not yet reliable.[10,11] In this study, only the subgrade Young's modulus is obtained by vibratory nondestructive test methods.

Furthermore, the subgrade is assumed to be infinitely thick. Previous work on the design for PCC pavements incorporates a stiff layer 20 ft below the pavement surface.[12] The present study found it unnecessary to incorporate a stiff layer at some arbitrary depth, so the computer programs SUBE and PAVEVAL assume a homogeneous subgrade.

The computer program PAVEVAL requires aircraft characteristics data as well as pavement parameters to calculate the allowable load-carrying capacity and the required overlay thickness of a pavement. These data include the load on one main gear wheel, the total number of main gear wheels, the tire contact area, and the wheel spacings.

PAVEMENT LAYER THICKNESSES

The pavement layer thicknesses are obtained from construction drawings or from measurements of core samples and thicknesses in core holes in the existing pavement if no construction records are available.

POISSON'S RATIO

The Poisson's ratio of the wearing surface and base and subbase courses was chosen according to the rules $\nu = 0.2$ for PCC pavements, $\nu = 0.3$ for AC pavements and AC base materials, and $\nu = 0.35$ for all other base and subbase materials. The Poisson's ratio for all subgrade

soils is taken to be $\nu = 0.35$. Different choices for these variables can be made according to the type of materials present.

YOUNG'S MODULUS

The Young's modulus of the PCC wearing surface of the PCC pavement is taken to be 4.0×10^6 psi. The temperature-dependent Young's modulus of AC pavements and AC base materials is obtained from Figure 3, corresponding to the pavement surface temperature at the time of the vibratory nondestructive testing. The temperature-dependent Young's modulus value is entered into the computer program SUBE to determine the subgrade Young's modulus. In this study, a value of $E = 450{,}000$ psi for AC pavements and AC base materials (corresponding to a yearly average temperature of 70°F) is entered into the computer program PAVEVAL that is to be compared with the conventional methods. However, in actual practice, a value of the AC Young's modulus is selected from the curve in Figure 3, representing the appropriate seasonal temperature.

The values of the Young's modulus of granular base and subbase materials can be estimated from the structure and composition of these materials. For instance, the laboratory resilient modulus test gives at least approximate values of the Young's modulus and the Poisson's ratio of these materials.[13-15] A reasonable estimate of the values of Young's modulus of base and subbase materials can be obtained from Table 1. The subgrade Young's modulus that is entered into the computer program PAVEVAL is the Young's modulus value predicted by the computer program SUBE.

If the base and subbase materials are completely unknown, it is possible to use a trial-and-error procedure to obtain the values for the Young's moduli of these materials and the value of the subgrade Young's modulus by using the nonlinear computer program SUBE and the curves in Figure 4.

The procedure for estimating the values of the Young's moduli of the base and subbase courses is as follows:

a. Select a trial value of the subgrade Young's modulus.
b. Use Figure 4 to obtain the Young's moduli of the base and subbase courses.

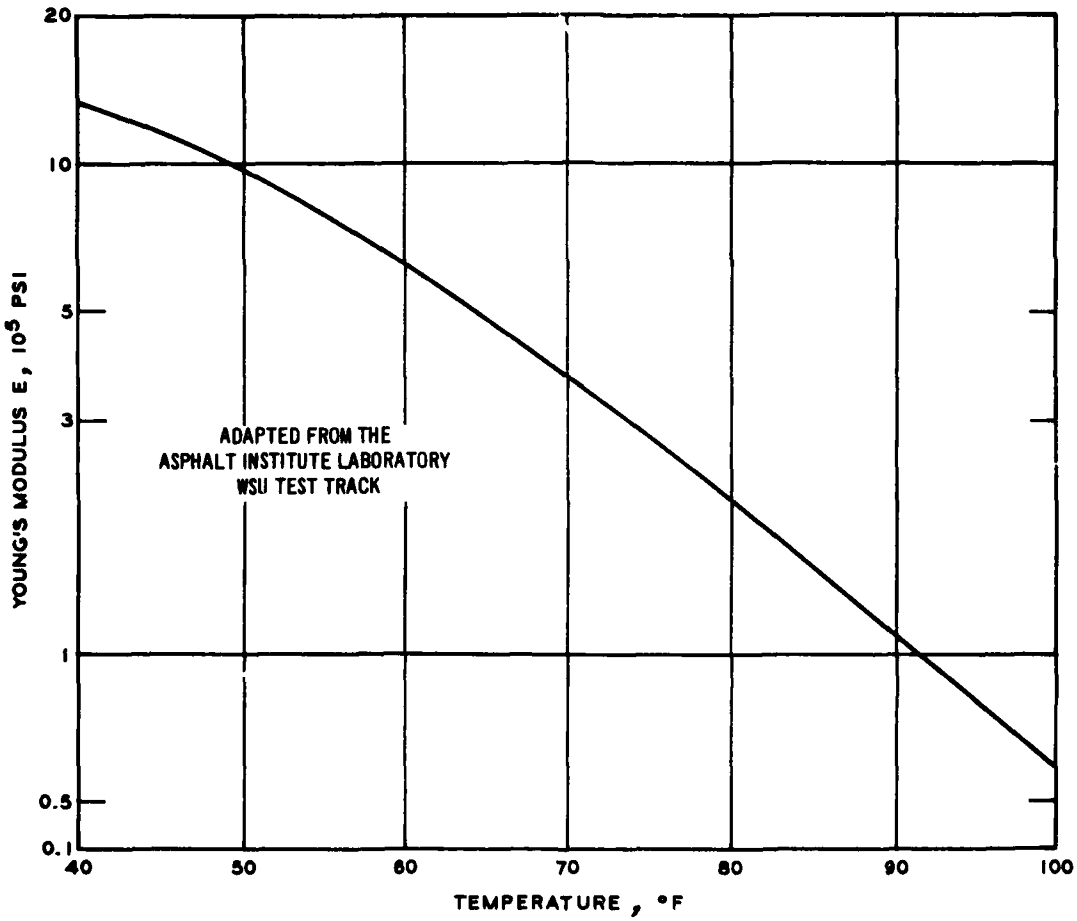


Figure 3. Assumed temperature dependence of Young's modulus of AC pavements and AC base materials

Table 1. Young's Modulus and Poisson's Ratio of Base and Subbase Materials

Material	Description	Assigned Value of Young's Modulus 10^3 psi	Assigned Value of Poisson's Ratio
Crushed limestone	Crushed limestone	80	0.35
GW	Well-graded gravel	60	0.35
GW-GM	GW and silty gravel	50	0.35
GP	Poorly graded gravel	40	0.35
GP-GC	GP and clayey gravels	35	0.35
SP	Poorly graded sand	30	0.35
SM	Silty sands, sand silt mixtures	30	0.35
SC	Clayey sands, sand clay	30	0.35
Black base	Mineral aggregate and bituminous material	Temperature dependent	0.30

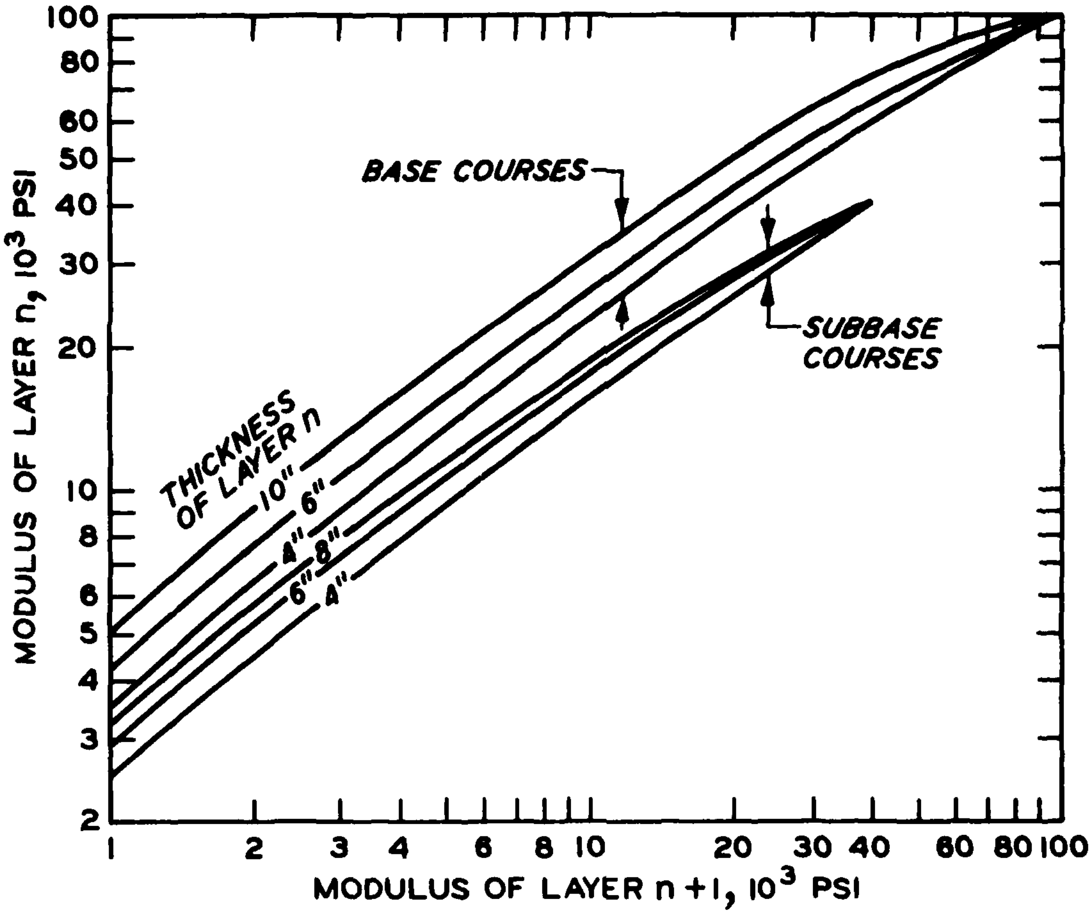


Figure 4. Relationship between the Young's modulus of layer n and the Young's modulus of layer n + 1 for various thicknesses of layer n

c. Place these trial values of the Young's moduli of the base and subbase courses (along with the measured DSM) into the computer program SUBE and get a new value of the subgrade Young's modulus.

d. Use the new value of subgrade Young's modulus to get new values of the Young's moduli of the base and subbase courses from Figure 4.

e. Repeat the procedure to the accuracy desired.

FLEXURAL STRENGTH

The flexural strengths of the PCC material of wearing surfaces can be measured in the laboratory on specimens cored from the PCC

pavement.[2] Splitting tensile tests are conducted, and the results are converted to flexural strengths. The approximate range of variation of the flexural strength of the PCC material found in the wearing surface of PCC pavements is $700 < R < 1020$ psi.[1]

AIRCRAFT CHARACTERISTICS

Pavement evaluation and overlay design procedures must include characteristics of the types of aircraft that operate at an airport. Basic aircraft data must be entered into the PAVEVAL computer program in order to calculate the allowable load-carrying capacity and the required overlay thickness for a pavement. The required aircraft data include the load on one wheel, the tire contact area, the total number of main gear wheels, and the transverse and longitudinal wheel spacings. Table 2 gives the required data for several aircraft in common use.

The load on one wheel used in Table 2 takes into consideration the assumption that 5 percent of the gross aircraft weight is supported by the nose wheel. The load on one wheel is therefore given by: gross weight × 0.95/number of main gear wheels. The operating load on a single wheel is used as the input load in the computer program PAVEVAL to calculate the overlay thickness for PCC and AC pavements. PAVEVAL automatically does the multiple-wheel calculation for the wheel configuration specified by the user.

Table 2. Aircraft Data

Aircraft Gear Configuration or Model Designation	Typical Gross Weight kips	Tire Contact Area sq in.	Total No. of Main Gear Wheels	Load on One Wheel kips	Transverse Wheel Spacing in.	Longitudinal Wheel Spacing in.
Single-wheel	30	190	2	14.25	--	--
Single-wheel	45	237	2	21.38	--	--
Single-wheel	60	271	2	28.50	--	--
Single-wheel	75	297	2	35.63	--	--
Dual-wheel	50	148	4	11.88	--	--
Dual-wheel	75	162	4	17.81	21	--
Dual-wheel	100	170	4	23.75	23	--
Dual-wheel	150	222	4	35.63	26	--
Dual-wheel	200	237	4	47.50	30	--
Dual-tandem	100	99	8	23.75	20	45
Dual-tandem	150	127	8	17.81	20	45
Dual-tandem	200	148	8	23.75	21	46
Dual-tandem	300	198	8	35.63	26	51
Dual-tandem	400	237	8	47.50	30	55
Boeing 727	173	210	4	41.09	34	--
DC-8-63F	358	220	8	42.51	32	55
Boeing 747	778	204	16	46.19	44	58
DC-10-10	433	294	8	51.42	54	64
DC-10-30	558	331	10	53.01	54	64
L-1011	428	282	8	50.83	52	70
Concorde	389	247	8	46.19	26.72	65.7
Boeing 737	111	174	8	13.18	30	--
Lockheed Electra	113	182	4	26.84	26	--
DC-9	115	165	4	27.31	25	--
Convair 880	188	152	8	22.33	22.5	45
Boeing 720	235	188	8	27.91	32	49
Boeing 707	336	218	8	39.90	34	56

LIMITING STRESS AND STRAIN CONDITIONS

As indicated previously, the failure of AC and PCC pavements can be related to limiting strain and stress conditions, respectively. Limiting stress and strain conditions are important for pavement evaluation because they relate the strain in the AC layer and the subgrade of AC pavements, and the stress in the PCC layer of PCC pavements, to the allowable load-carrying capacity and the required overlay thickness of a pavement. Because the stress and strain at any point in a pavement depends on the pavement structure, the limiting stress and strain concept relates the allowable load-carrying capacity and the required overlay thickness to the pavement structure, i.e., to the thickness and the elastic moduli of the pavement layers.

AC PAVEMENTS

The soil of the subgrade will undergo excessive plastic flow under repetitive loads if the repeated vertical strain at the top of the subgrade exceeds a limiting value.[9] The limiting value of the vertical strain at the top of the subgrade depends on the number of strain repetitions and on the value of the Young's modulus of the soil in the subgrade. Figure 5 gives the limiting vertical strain ϵ_{VL} as a function of the subgrade Young's modulus for 1,200, 6,000, and 25,000 annual strain repetitions.[9] The curves in Figure 5 are assumed to be valid for all types of subgrade soil and for single- and multiple-wheel loadings.

Figure 6 gives the limiting vertical strain at the top of the subgrade of AC pavements in terms of the total number of load repetitions independent of the value of the subgrade Young's modulus.[9] A straight-line representation of the data in Figure 6 can be written as

$$\log \epsilon_{VL} = A \log N + B \qquad (1)$$

where

ϵ_{VL} = limiting vertical strain at the top of the subgrade

N = total number of load repetitions to failure

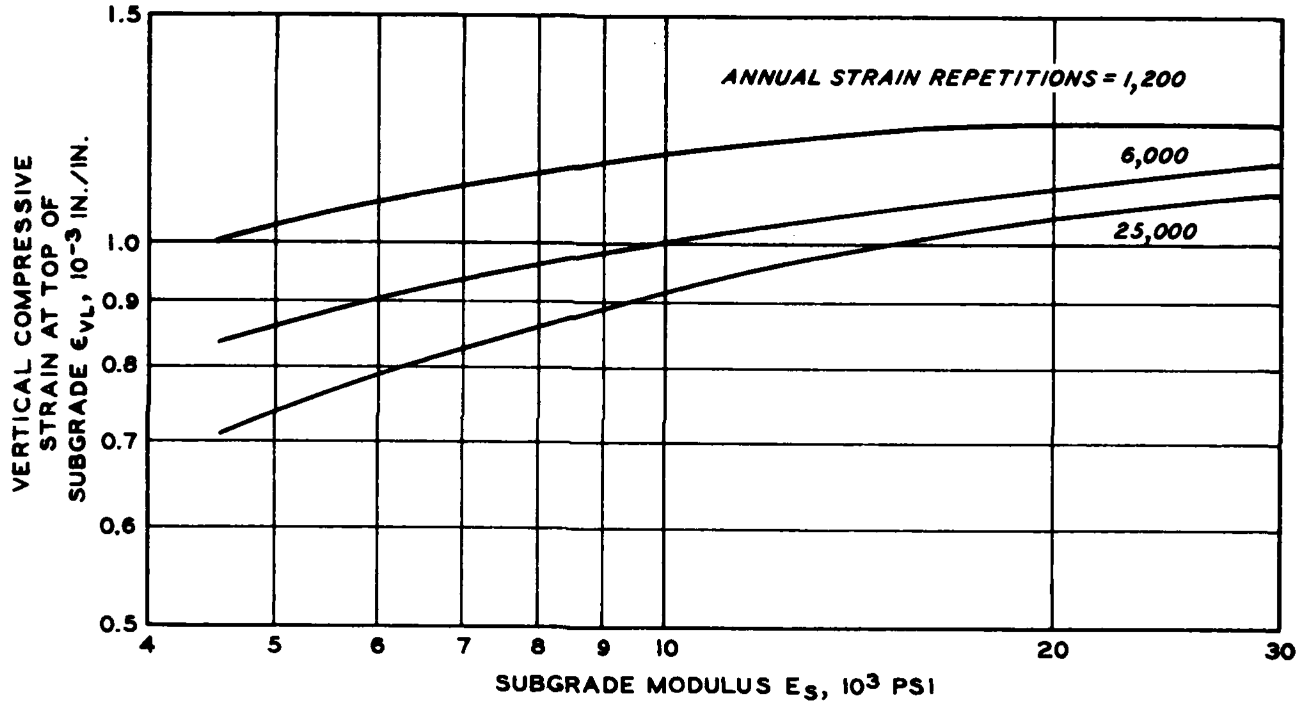


Figure 5. Limiting subgrade strain in terms of the subgrade Young's modulus for conventional AC pavements

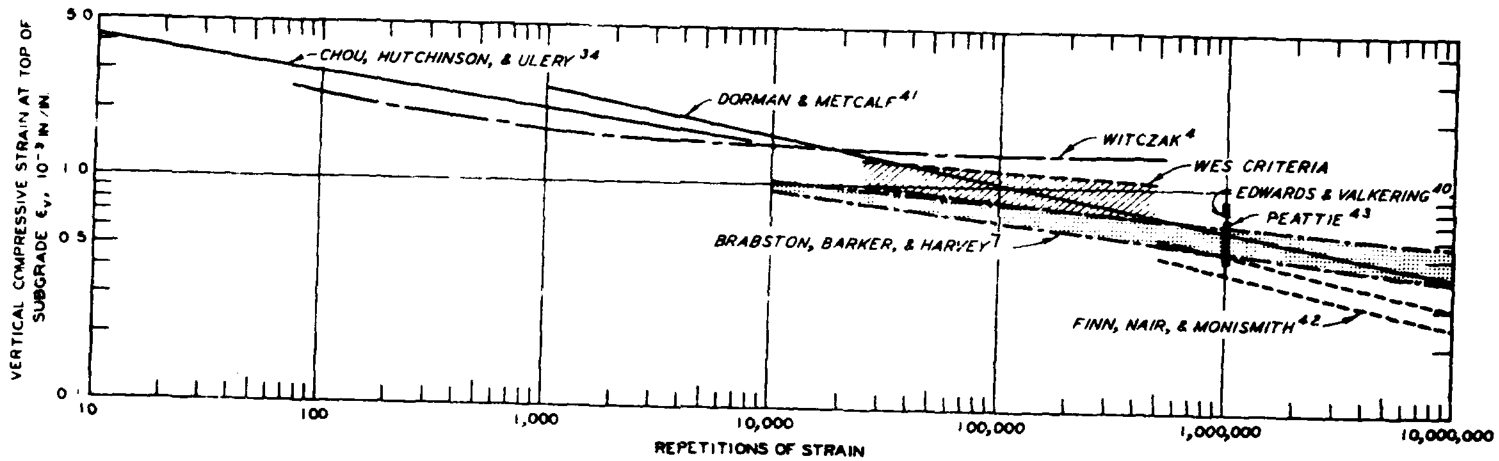


Figure 6. Comparison of subgrade strain criteria (Reference 9)

A best-fit curve through all the data in Figure 6 gives the values, A = -0.162 and B = -2.22 , for the coefficients.

Figure 7 gives the limiting value of the tensile strain ε_{RL} at the bottom of the AC layer.[9] The limiting vertical strain in the subgrade is found to be the controlling condition in most AC pavements, and for all cases considered in this study, it was found that the limiting vertical strain in the subgrade overshadowed the limiting tensile strain in the AC layer.

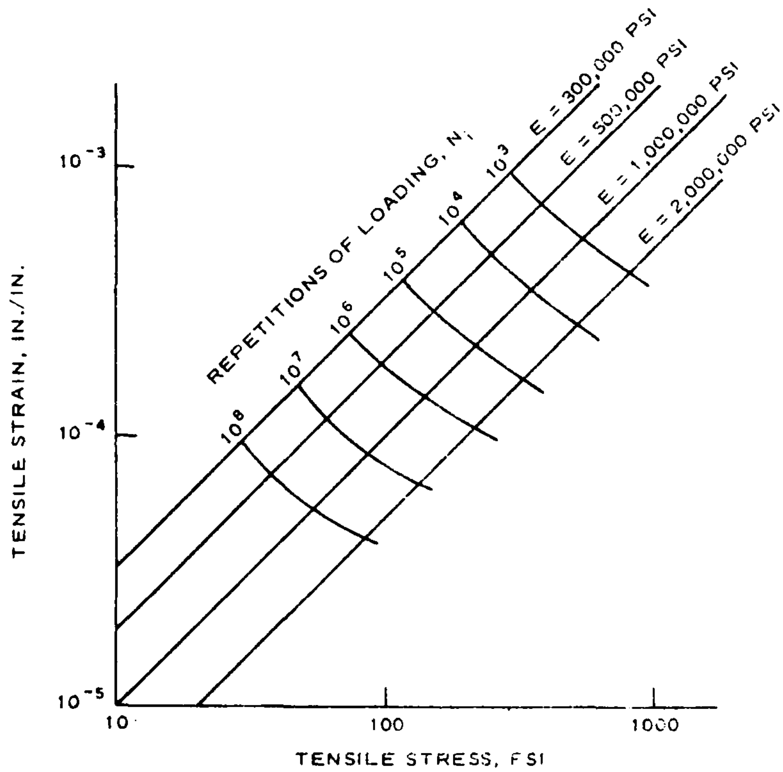


Figure 7. Limiting tensile strain at the bottom of the AC wearing surface

PCC PAVEMENTS

It is assumed that a load applied to the surface of a PCC pavement produces a maximum tensile stress at the bottom of the PCC layer. Further, cracking is assumed to occur first at the bottom of this layer. These are poor assumptions since failure often occurs at

the joints, and the location of the load or of curling conditions is not considered. This incipient cracking will probably be the onset of failure in a PCC pavement; it will begin in the PCC layer when the applied tensile stress at the bottom of this layer exceeds a limiting value of tensile stress.[8,9] The limiting tensile stress is expressed in terms of the number of load (stress) repetitions and in terms of the flexural strength of the PCC layer as

$$\sigma_{RL} = \frac{R}{A + B \log (COV)} \tag{2}$$

where

σ_{RL} = limiting value of tensile stress, psi
R = flexural strength, psi
A = 0.58901
B = 0.35486
COV = number of coverages

This expression is assumed to be valid whether the stress in the PCC layer is produced by a single- or a multiple-wheel loading.

The number of coverages is related to the number of load repetitions by a factor that depends on the type of aircraft operating on a runway. A coverage refers to a load covering the full width of the traffic lane and thus must be related to the number of repetitions of a particular gear configuration. The connection is made through a pass-to-coverage ratio FAC, which is given by $FAC = N/COV$ where N = number of load repetitions. The limiting radial tensile stress given by Equation 2 can be expressed in terms of the number of load repetitions if the pass-to-coverage ratio is specified. Each gear configuration is associated with a unique value of the pass-to-coverage ratio. Table 3 gives values of the pass-to-coverage ratio for various gear configurations.[16]

Table 3. Pass-to-Coverage Ratios

Aircraft Wheel Configuration Type	Ratio (FAC)
Single	5.18
Dual	3.48
Dual-tandem	3.68
L-1011	3.62
B-747	3.70
DC-10-10	3.64
DC-10-30	3.38
DC-8	3.14

SUBGRADE YOUNG'S MODULUS DETERMINED BY VIBRATORY NONDESTRUCTIVE TESTING OF PAVEMENTS

GENERAL CONSIDERATIONS

The basic purpose of the vibratory nondestructive testing of pavements is to supply pavement parameters for the layered elastic theoretical calculation of the allowable load-carrying capacity and the required overlay thickness of a pavement. The layered elastic model of pavements requires the Young's modulus, the layer thickness, and the Poisson's ratio of the subgrade and pavement layers to be known. The elastic moduli of the pavement layers are estimated by various means, and only the subgrade Young's modulus is determined from vibratory nondestructive test results. The subgrade Young's modulus calculated by means of the computer program SUBE serves as an input pavement parameter for the layered elastic theory computer program PAVEVAL that is used for pavement evaluation and overlay design.

Pavements have been noted to behave nonlinearly under dynamic loadings.[1,3] A nonlinear dynamic layered elastic theory and the computer program SUBE have been developed that determine the subgrade Young's modulus directly from the vibratory nondestructive test data measured at the pavement surface.[3,4] The input pavement parameters for this dynamic elastic theory are the elastic modulus and the thickness of each pavement layer and the Poisson's ratio of the subgrade. The input from the vibratory nondestructive test data is the dynamic load-deflection curve measured at the surface of a pavement by the WES 16-kip vibrator.

The BISAR computer program is used for the design of PCC and AC pavements[9,12] and has been modified for pavement evaluation and designated PAVEVAL.

VIBRATORY NONDESTRUCTIVE TEST DATA

The WES 16-kip vibrator applies a static load of 16 kips to the pavement surface and a dynamic load up to 15 kips at frequencies

ranging from 5 to 100 Hz. Both static and dynamic loads are applied to the pavement surface through a circular 18-in.-diam baseplate.

Four types of vibratory nondestructive tests are generally performed on pavements:

- a. Dynamic load-deflection curves that show the dynamic deflection of the pavement surface as a function of the applied load for a fixed frequency of 15 Hz.
- b. Frequency response spectrum that shows the dynamic deflection as a function of frequency for a fixed dynamic load.
- c. Deflection basin measurements.
- d. Rayleigh surface wave dispersion curves that show phase velocity versus wavelength (or frequency).

Only test a above is conducted and used in the method reported to determine the subgrade Young's modulus.

Figure 8 presents a typical dynamic load-deflection curve measured at 15 Hz. The dynamic deflection of the pavement surface is a nonlinear function of the dynamic load applied to the pavement surface. The slope of the dynamic load-deflection curve (tangent modulus) is called the DSM. The numerical value of the DSM is generally obtained from the region of high dynamic loading. Because the dynamic load-deflection curves are nonlinear, a nonlinear dynamic theory is required for their description and to extract the value of the subgrade Young's modulus.[3,4]

NONLINEAR DYNAMIC THEORY OF PAVEMENT RESPONSE

The nonlinear dynamic theory of pavement response was developed to describe the dynamic load-deflection curves that are measured at the pavement surface and to predict the value of the subgrade Young's modulus from the vibratory nondestructive test measurements.[3,4] The nonlinear theory of pavement response develops and solves the equation of motion of a nonlinear oscillator and gives a theoretical expression for the dynamic deflection of the pavement surface in terms of the dynamic load applied to the pavement surface. The parameters that describe the nonlinear pavement response are related to the elastic moduli of the pavement layers and the subgrade.[3] For a specified

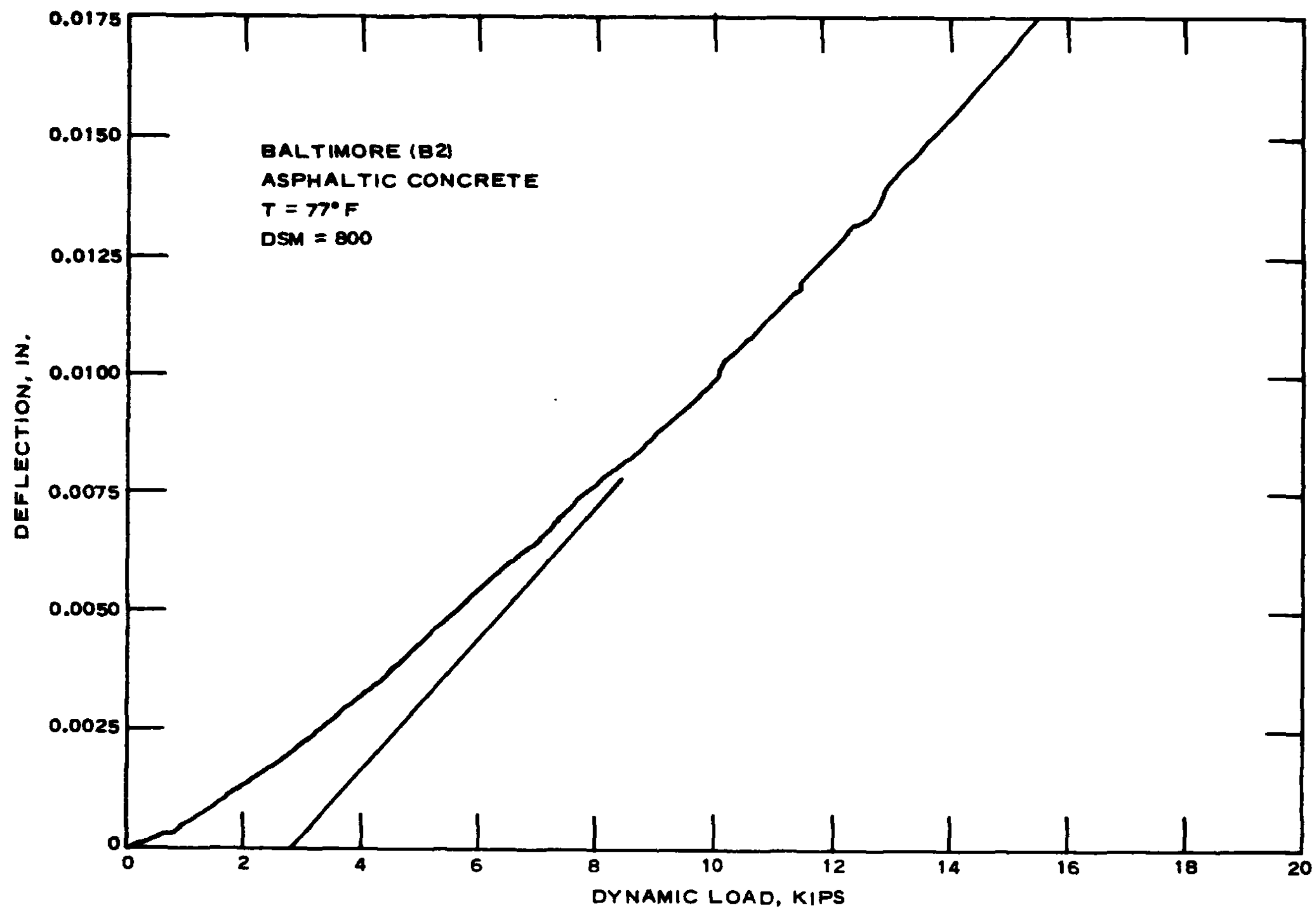


Figure 8. Typical dynamic load-deflection curve for the AC pavement

choice of the elastic moduli of the pavement layers and the Poisson's ratio of the subgrade, the value of the subgrade Young's modulus is obtained by requiring that the theoretically predicted dynamic load-deflection curve agree with the measured dynamic load-deflection curve.[3,4]

DYNAMIC PAVEMENT RESPONSE COMPUTER PROGRAM SUBE

The computer program SUBE calculates the value of the subgrade Young's modulus from input data taken from the measured dynamic load-deflection curves.[4] The pavement input parameters for SUBE include the Young's modulus, the Poisson's ratio, and the thickness of each pavement layer, as well as the Poisson's ratio of the subgrade. The computer input that is taken from vibratory nondestructive test data is the DSM value and a point-by-point description of the measured dynamic load-deflection curve. From the DSM value, SUBE calculates the effective mass, the damping constant, the finite depth of influence of the static stress-strain field, and all the other parameters that enter into the nonlinear theoretical model of pavement response.[4] SUBE iterates the value of the subgrade Young's modulus and determines the value of the subgrade Young's modulus that makes the theoretically predicted DSM value agree with the measured DSM value so that the theoretically predicted dynamic load-deflection curve will agree with the measured dynamic load-deflection curve. Figure 9 outlines the procedure.

NUMERICAL VALUES OF THE PREDICTED SUBGRADE YOUNG'S MODULUS

Tables 4 and 5 show the pavement structures for which dynamic load-deflection curves were measured. These tables also present the values of the elastic moduli of the pavement layers that were used in the computer program SUBE to predict the values of the subgrade Young's modulus. Figure 10 shows a comparison of the subgrade modulus values predicted by the nonlinear dynamic response theory through SUBE and the subgrade modulus values predicted by the formula $E_s = 1500$ CBR.[17] The predicted subgrade Young's modulus values depend on the choice of the values of the Young's moduli of the pavement layers. No CBR or

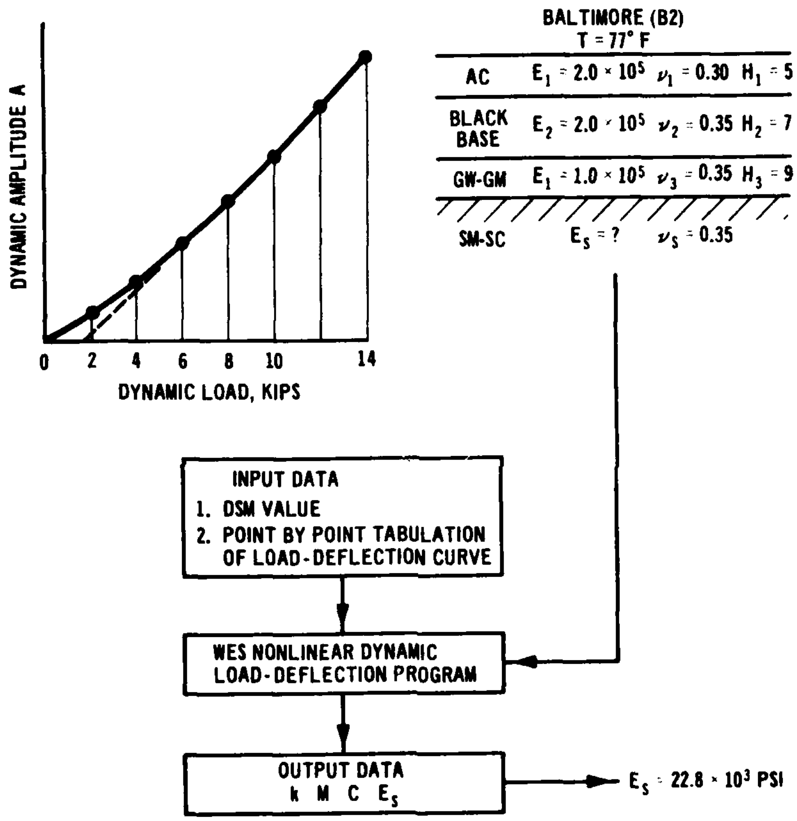


Figure 9. Procedure for obtaining the subgrade Young's modulus from the measured dynamic load-deflection curve

Table 4. AC Pavement Structures Investigated

Site	DSM kips/in.	Wearing Surface E_1 psi	Wearing Surface ν_1	Wearing Surface h_1 in.	Base E_2 psi	Base ν_2	Base h_2 in.	Subbase E_3 psi	Subbase ν_3	Subbase h_3 in.	Subgrade E_s WES Nonlinear psi	Subgrade ν_s	Subgrade CBR	Subgrade E_s 1500 CBR psi
WES-WEL area subgrade	300	--	--	--	--	--	--	--	--	--	12,300	0.35 Loess	8	12,000
WES hangar No. 4 subgrade	400	--	--	--	--	--	--	--	--	--	10,700	0.35 Lean + heavy clay	31	46,500
TETS-adjacent subgrade	320	--	--	--	--	--	--	--	--	--	27,000	0.35 Lean clay	14	21,000
TETS-poorhouse subgrade	300	--	--	--	--	--	--	--	--	--	15,900	0.35 Lean clay	X	X
TETS-adjacent subgrade	450	--	--	--	--	--	--	--	--	--	13,000	0.35 Lean clay	8	12,000
B2A asphaltic concrete	700	230,000 AC T = 77°F	0.3	5	230,000 AC base T = 77°F	0.35	7	32,000 GW-GM	0.35	9	25,000	0.35 SM-SC	14	21,000
N18 asphaltic concrete	770	1,400,000 AC T = 39°F	0.3	3.25	34,000 GP	0.35	6.0	--	--	--	29,600	0.35	18	27,000
WES test area asphaltic concrete	780	100,000 AC T = 90°F	0.3	3.0	30,000 Crushed limestone	0.35	6.0	200,000 Lime-stabilized lean clay	0.35	24.0	6,700	0.35 Heavy clay	4	6,000
WI asphaltic concrete	860	180,000 AC T = 87°F	0.3	9.0	40,000 GW-GM	0.35	5.0	--	--	--	19,500	0.35 SP-SC	20	30,000
Alum Creek 123-11 asphaltic concrete	820	1,300,000 AC T = 41°F	0.3	7.8	--	--	--	--	--	--	19,400	0.35 GC	10	15,000

(Continued)

Table 4 (Concluded)

Site	DSM kips/in.	Wearing Surface E_1 psi	Wearing Surface ν_1	Wearing Surface h_1 in.	Base E_2 psi	Base ν_2	Base h_2 in.	Subbase E_3 psi	Subbase ν_3	Subbase h_3 in.	Subgrade E_s WES Nonlinear psi	Subgrade ν_s	Subgrade CBR	Subgrade E_s 1500 CBR psi
Alum Creek 123-2 asphaltic concrete	880	1,300,000 AC	0.3 T = 41°F	7.8	--	--	--	--	--	--	19,000	0.35 GC	10	15,000
N23A asphaltic concrete	980	1,300,000 AC	0.3 T = 41°F	3.0	1,300,000 AC base	0.3 T = 41°F	3.0	32,000	0.35 GP	7.0	28,000	0.35 SM	18	27,000
Alum Creek 123-23 asphaltic concrete	1000	1,300,000 AC	0.3 T = 41°F	7.6	--	--	--	--	--	--	28,000	0.35 GC	17	25,500
Alum Creek 123-32 asphaltic concrete	1230	1,300,000 AC	0.3 T = 41°F	7.6	--	--	--	--	--	--	21,000	0.35 GC	17	25,500
B3 asphaltic concrete	1680	700,000 AC	0.3 T = 58°F	5.5	700,000 AC base	0.3 T = 58°F	5.5	35,000	0.35 GW-GM	10.5	30,000	0.35 SP-SM	17	25,500
W2C asphaltic concrete	1940	300,000 AC	0.3 T = 73°F	12.0	60,000	0.35 GW-GM	6.0	--	--	--	35,000	0.35 SP-SC	20	30,000
P14A asphaltic concrete	2120	200,000 AC	0.3 T = 80°F	14.0	53,000	0.35 GP-GC	5.0	--	--	--	28,700	0.35 GP-GC	20	30,000
P13 asphaltic concrete	2780	320,000 AC	0.3 T = 73°F	15.0	60,000	0.35 GP-GC	5.0	--	--	--	35,000	0.35 GP-GC	20	30,000
B1 asphaltic concrete	3120	500,000 AC	0.3 T = 65°F	5.0	500,000 AC base	0.3 T = 65°F	12.5	28,000	0.35 SP-SM	6.0	21,000	0.35 SM	14	21,000
WES test area PCC	3500	4,000,000	0.2 PCC	15.0	200,000	0.3 AC base	6.0	--	--	--	6,800	0.35 Heavy clay	4	6,000

Table 5. Rigid Pavement Structures Investigated

Site	F psi	DSM kips/in.	Wearing Surface Classification	E_1 psi	ν_1	h_1 in.	Base Classification	E_2 psi	ν_2	h_2 in.	Subgrade Classification	E_s psi	ν_s
N7	810	1866	PCC	4,000,000	0.2	7.0	E-1(GW)	50,000	0.35	8.0	E-4(SP-SM)	20,000	0.35
N9	724	1920	PCC	4,000,000	0.2	10.0	E-1(GW)	40,000	0.35	8.0	E-4(SP-SM)	11,000	0.35
N11	894	1050	PCC	4,000,000	0.2	7.0	E-7(ML-CL)	16,000	0.35	8.0	E-4(SP-SM)	14,100	0.35
N8	880	1600	PCC	4,000,000	0.2	7.0	E-1(GW)	50,000	0.35	8.0	E4(SP-SM)	21,500	0.35
J5	966	1760	PCC	4,000,000	0.2	8.75	Soil-cement	18,000	0.35	8.0	E7(CL)	14,000	0.35
J3	740	2060	PCC	4,000,000	0.2	7.5	Soil-cement	25,000	0.35	4.75	E7(CL)	28,000	0.35
I20	984	2300	PCC	4,000,000	0.2	10.0	Gravelly sand	28,000	0.35	8.5	E4(SP-SM)	9,000	0.35
N15	864	2640	PCC	4,000,000	0.2	10.0	E-7(ML-CL)	4,500	0.35	9.0	E4(SP-SM)	12,000	0.35
I13	870	3240	PCC	4,000,000	0.2	12.0	Gravelly sand	21,000	0.35	6.0	E6(SP-SC)	17,800	0.35
J1	754	2940	PCC	4,000,000	0.2	9.5	Soil-cement	15,000	0.35	6.25	E-7(CL)	16,000	0.35
I8	1003	3100	PCC	4,000,000	0.2	12.0	Gravelly sand	33,000	0.35	11.5	E-4(SP-SM)	23,300	0.35
H7	929	3480	PCC	4,000,000	0.2	12.0	E-5(SP-SM)	60,000	0.35	5.5	E-6(ML)	23,200	0.35
H1	868	3520	PCC	4,000,000	0.2	14.0	Soil-cement	16,000	0.35	9.0	E-6(ML)	17,200	0.35
H11	894	3700	PCC	4,000,000	0.2	12.5	E-5(SP-SM)	82,000	0.35	11.0	E-6(ML)	21,000	0.35
H3	929	4363	PCC	4,000,000	0.2	12.0	Soil-cement	60,000	0.35	8.0	E-6(ML)	15,000	0.35

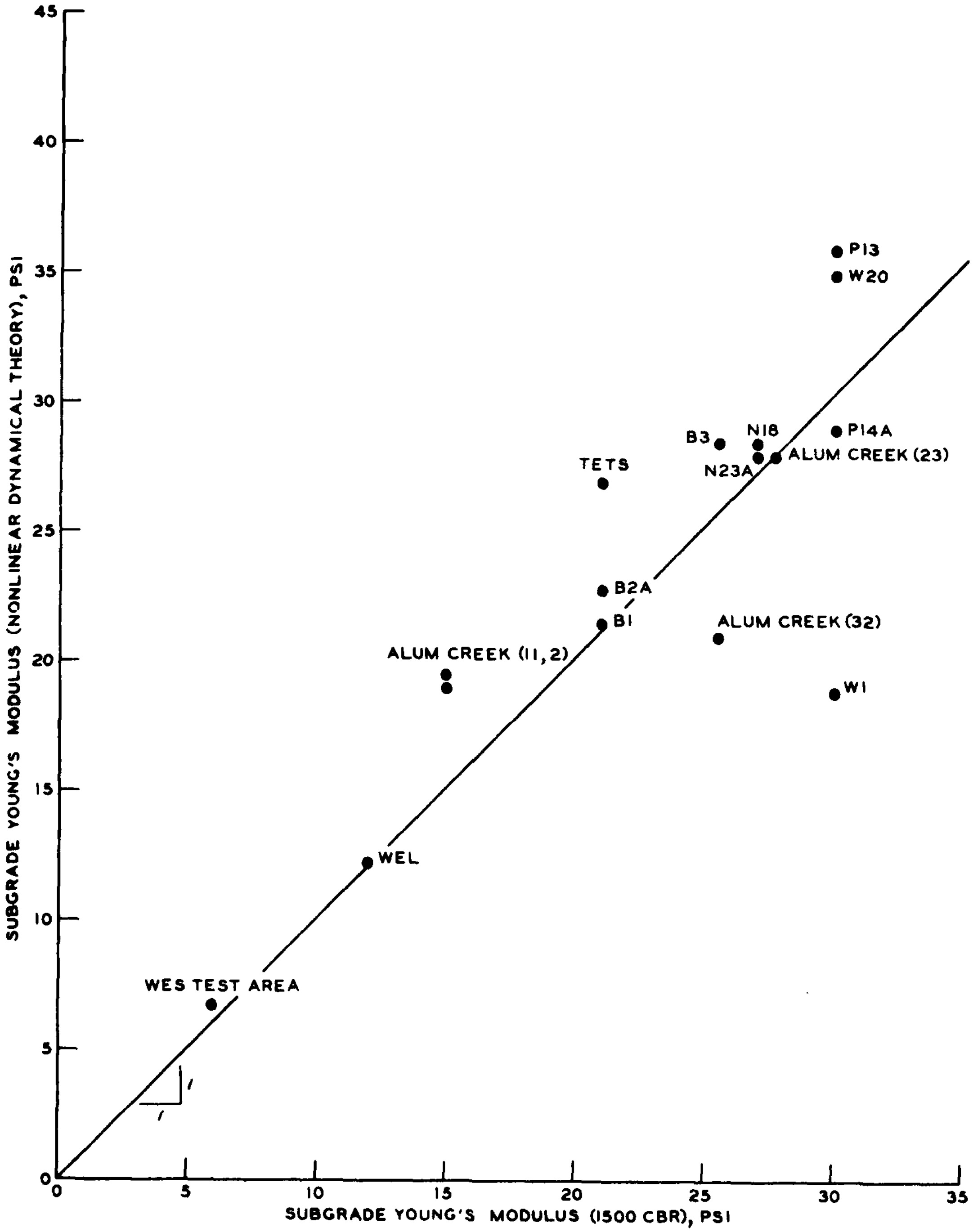


Figure 10. Comparison of subgrade Young's modulus values predicted by the nonlinear dynamic theory computer program SUBE and by the wave propagation formula $E_s = 1500$ CBR

coefficient of subgrade reaction values were measured for the subgrade at the PCC pavement sites.

The formula $E_s = 1500$ CBR, where E_s represents the subgrade Young's modulus, is obtained as a best-fit straight line through data points for which there was considerable scatter. Therefore, this relationship should be considered to be approximately true, and many deviations from the rule may occur according to the type of materials present and the extreme values of the CBR that may be encountered. The nonlinear dynamic theory of pavement response and the associated computer program SUBE were developed to predict values of the subgrade Young's modulus that are in reasonable agreement with the predictions of the formula $E_s = 1500$ CBR. The predicted values of the subgrade Young's modulus can also be compared with laboratory resilient modulus measurements, but this comparison was not made in this study.

Some studies of the sensitivity of the predicted value of the subgrade Young's modulus on the choice of the value of the elastic moduli of the pavement layers have been conducted. Figure 11 shows the dependence of the predicted subgrade modulus values on the values of the Young's moduli of the pavement layers at a pavement site where the DSM value has been measured. The basic pavement structure about which the Young's modulus value of each pavement layer was varied one at a time is as follows:

E_1 = 200,000 psi h_1 = 5.0 in.
E_2 = 80,000 psi h_2 = 7.0 in.
E_3 = 40,000 psi h_3 = 9.0 in.

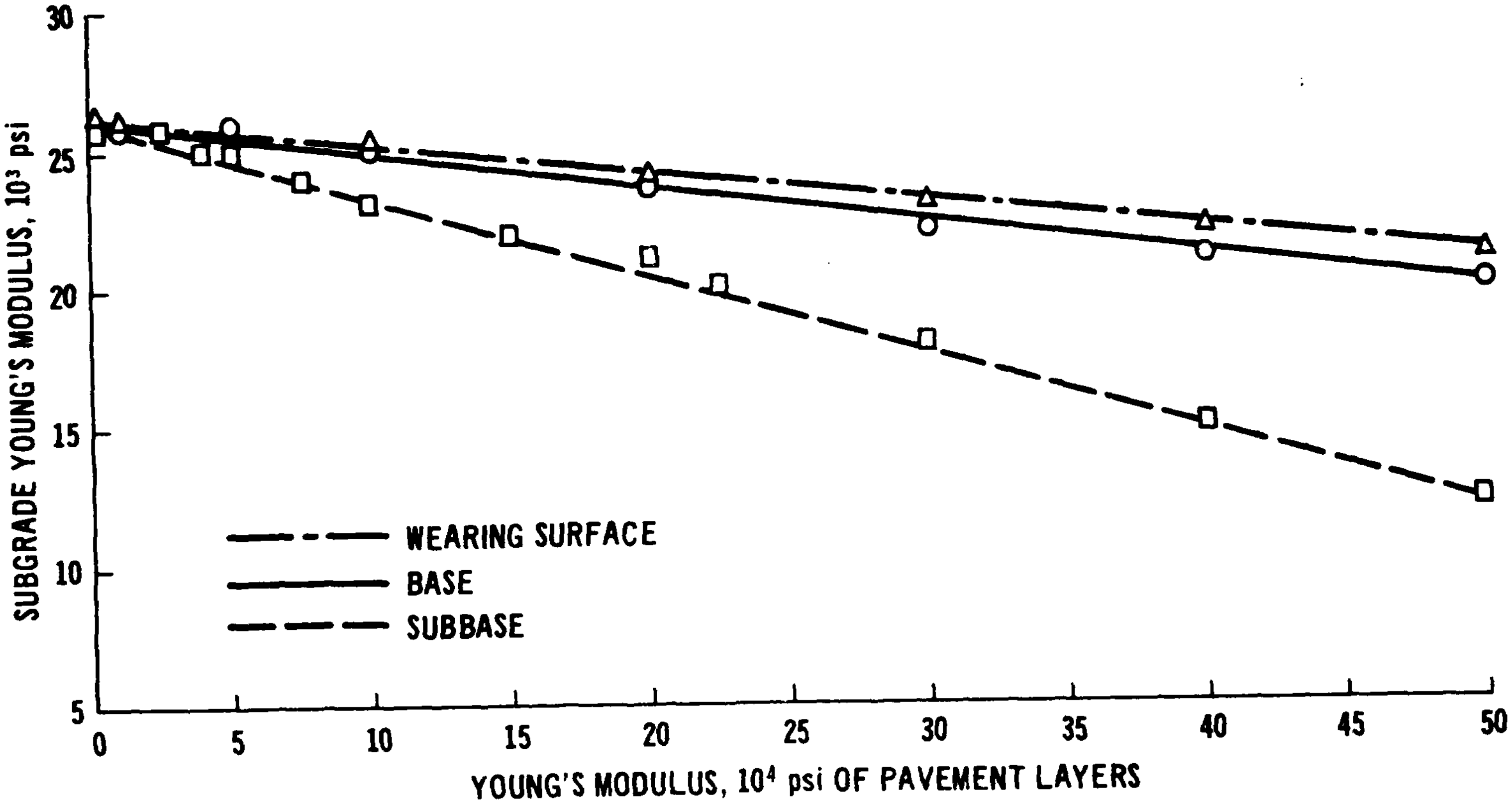


Figure 11. Dependence of the subgrade Young's modulus values predicted by the computer program SUBE on the chosen values of the Young's modulus of the wearing surface and the base and subbase courses

LAYERED ELASTIC THEORY CALCULATION OF THE STRESS AND THE STRAIN IN PAVEMENTS

GENERAL CONSIDERATIONS

The determination of the load-carrying capacity of a pavement and the overlay thickness required to upgrade a pavement entails the calculation of the vertical compressive strain at the top of the subgrade or the tensile strain at the bottom of the AC layer for AC pavements, and the tensile stress at the bottom of the PCC layer for PCC pavements. The calculation of the stress and the strain at points in the pavement and the subgrade is accomplished by modeling the pavement and the subgrade as a semi-infinite layered elastic halfspace for which each layer is described by a Young's modulus, a Poisson's ratio, and a thickness. The layered elastic theory connects the allowable load at the pavement surface and the required overlay thickness for PCC pavements with the limiting values of tensile stress at the bottom of the PCC layer; and for AC pavements, with the compressive vertical strain at the top of the subgrade or the tensile strain at the bottom of the AC layer. The BISAR computer program is used to implement the basic layered elastic theory.

The input parameters for the layered elastic theoretical model of a pavement are the elastic moduli and the thickness of each pavement layer. As discussed previously, the subgrade Young's modulus can be obtained from vibratory nondestructive tests conducted at the surface of the pavement, and the Young's modulus of the wearing surface and the base and subbase courses can be obtained from a classification of the material or from the measured CBR. The thicknesses of the pavement layers are obtained from construction specifications or from measurements in the field. Therefore, all of the parameters required by the layered elastic theory are available for pavement structures.

BISAR COMPUTER PROGRAM

The BISAR computer program was developed by the Shell Oil Company for pavement applications. This computer program calculates the stress

and the strain at any point in the pavement or the subgrade due to a loading at the pavement surface. Particle displacements, stresses, and strains are obtained by numerical integration.

Boundary conditions between the pavement layers may be taken to be rough or smooth. For the rough condition, the radial and tangential stresses and strains are continuous across the layer interfaces. For the smooth condition, the radial and tangential stresses and strains are not continuous across the interface. For PCC pavements, the interface between the PCC wearing surface and the base is assumed to be smooth, while all other interfaces are taken to be rough. For AC pavements, all interfaces are assumed to be rough.

Each pavement layer is characterized by a thickness, a Poisson's ratio, and a Young's modulus. Therefore, three parameters must be specified for each pavement layer. The value of the surface load and the size of the circular loaded area must be specified. The coordinates of the point in the pavement where the stress and the strain are to be calculated must also be specified. The BISAR computer program has the capability of calculating the stress and the strain in the pavement when more than one load is applied to the pavement surface.

The BISAR computer program is modified to calculate the overlay thickness required to upgrade a pavement and the allowable load-carrying capacity of a pavement. The modification consists of an iterative procedure to match the calculated stress and strain with specified limiting values of the stress and the strain. The resulting computer program is called PAVEVAL.

ALLOWABLE LOAD-CARRYING CAPACITY AND REQUIRED OVERLAY THICKNESS OF PAVEMENTS

GENERAL CONSIDERATIONS

The allowable load-carrying capacity and the required overlay thickness of a pavement is related to the pavement structure. The layered elastic theory relates the allowable load-carrying capacity and the required overlay thickness to the pavement structure as represented by the elastic modulus and the thickness of each pavement layer. The following paragraphs describe the layered elastic method of pavement evaluation and overlay design.

CHOICE OF ELASTIC MODULI FOR PAVEVAL COMPUTER PROGRAM

The value of the subgrade Young's modulus that is used in the PAVEVAL computer program to calculate the allowable load-carrying capacity and the required overlay thickness of a pavement is obtained by using the computer program SUBE to analyze the dynamic load-deflection curves measured at a pavement site. The choice of the elastic moduli of the pavement layers that are entered into PAVEVAL are the same as those selected for SUBE with the exception that the Young's modulus of AC pavements and AC base materials is chosen always to have the value $E = 450{,}000$ psi in PAVEVAL. This value of the Young's modulus is obtained from Figure 3, corresponding to an assumed average yearly pavement temperature of 70°F.

The values of the Young's modulus of AC pavements and AC base materials that are used in the computer program SUBE to calculate the subgrade Young's modulus are obtained from Figure 3 for a temperature equal to the pavement temperature at the time of the measurement of the dynamic load-deflection curves.

SINGLE-WHEEL LOADING

A point in the pavement and the subgrade is designated to have coordinates x,y,z, where x and y describe the horizontal plane and z measures the depth beneath the pavement surface (Figure 12). The

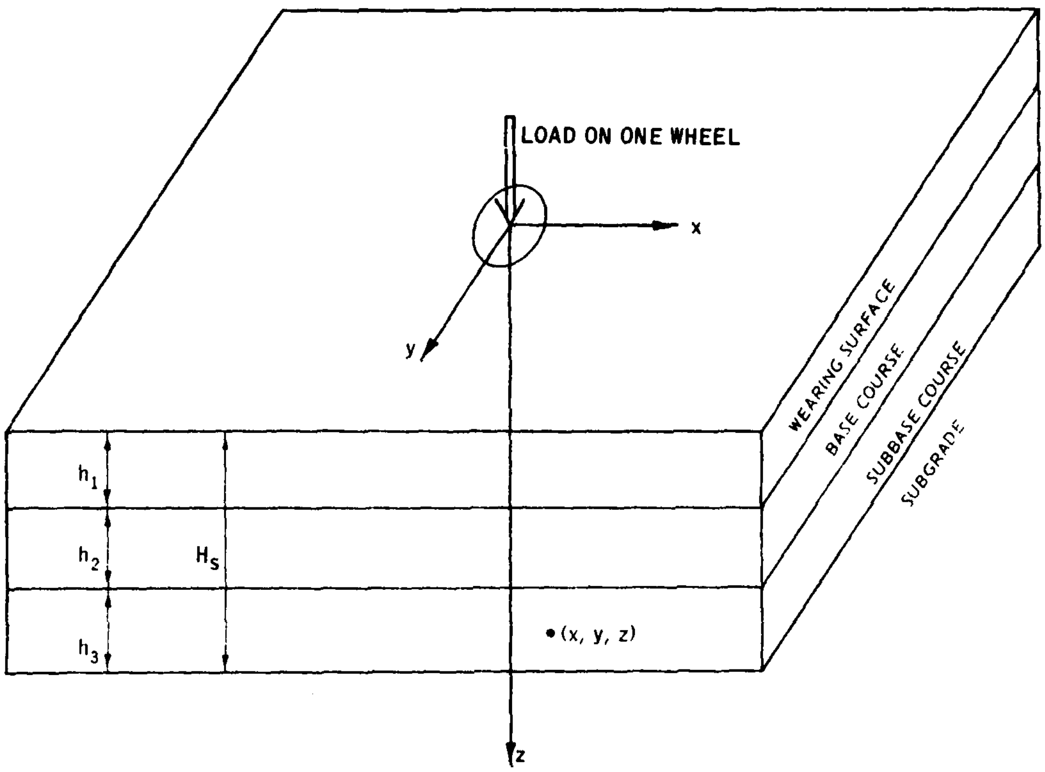
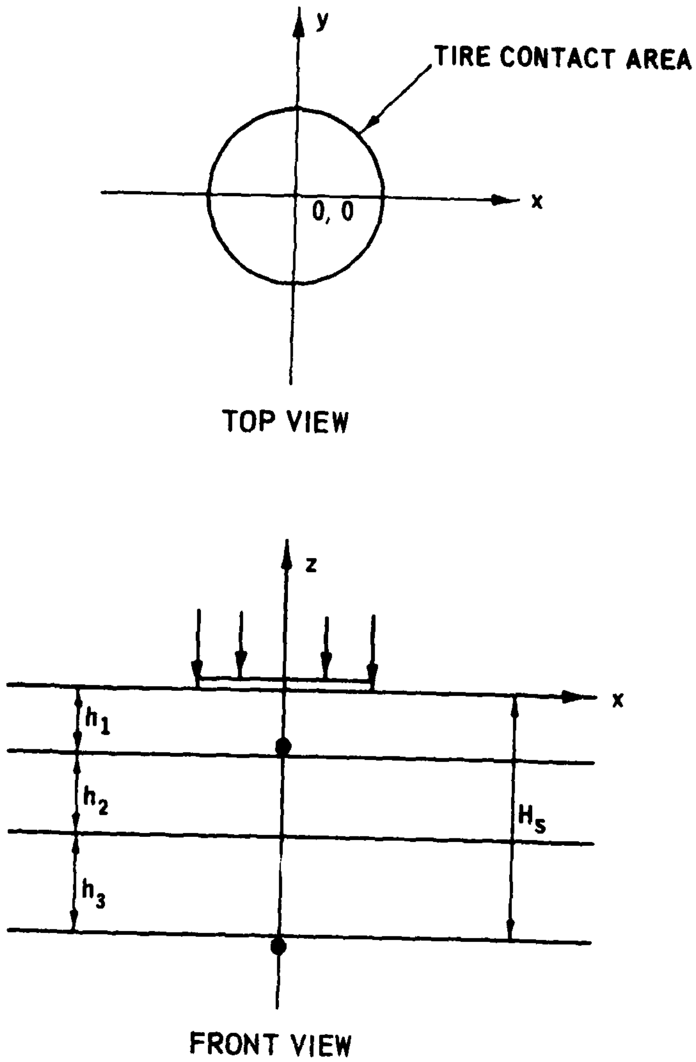


Figure 12. Coordinate system chosen for the layered elastic theory of pavements

vertical strain ε_V, the tensile strain ε_R, and the radial stress σ_R at a point in the pavement are functions of the coordinates of the point in the manner $\varepsilon_V = \varepsilon_V(x,y,z)$, $\varepsilon_R = \varepsilon_R(x,y,z)$, $\sigma_R = \sigma_R(x,y,z)$. The maximum values of the stress and the strain in the pavement and the subgrade occur directly beneath the single-wheel load, so that if a coordinate system is chosen whose origin is at the center of the single-wheel load, as shown in Figures 12 and 13, the vertical strain at the top of the subgrade, the radial strain at the bottom of the AC layer, and the tensile stress at the bottom of the PCC layer are represented by



$$\left.\begin{aligned} \epsilon_V &= \epsilon_V\,(0,\ 0,\ H_s) \\ \epsilon_R &= \epsilon_R\,(0,\ 0,\ h_1) \end{aligned}\right\} \quad \text{AC PAVEMENT}$$

$$\sigma_R = \sigma_R\,(0,\ 0,\ h_1) \qquad \text{PCC PAVEMENT}$$

Figure 13. Calculation of the stress and the strain for a single-wheel loading

$$\left.\begin{aligned}\epsilon_V &= \epsilon_V\ (0,0,H_s)\\ \epsilon_R &= \epsilon_R\ (0,0,h_1)\end{aligned}\right\} \qquad (3)$$

$$\sigma_R = \sigma_R\ (0,0,h_1) \qquad (4)$$

where

H_s = depth to the top of the subgrade of an AC pavement

h_1 = thickness of a PCC or an AC wearing surface

The conditions that determine the allowable load-carrying capacity and the required overlay thickness are

$$\left.\begin{aligned}\epsilon_R &= \epsilon_{RL}\\ \epsilon_V &= \epsilon_{VL}\end{aligned}\right\} \quad \text{AC pavement} \qquad (5)$$

$$\sigma_R = \sigma_{RL} \quad \text{PCC pavement} \qquad (6)$$

where

ϵ_V, ϵ_R and σ_R = values from Equations 3 and 4, respectively

ϵ_{VL} = limiting strain from Figures 5 and 6

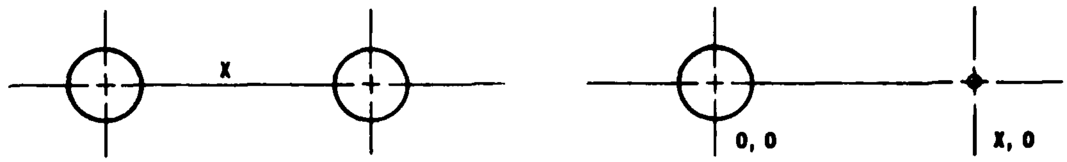
ϵ_{RL} = tensile strain from Figure 7

σ_{RL} = limiting stress from Equation 2

MULTIPLE-WHEEL LOADING

Actual aircraft loadings on a pavement occur through two or more wheels in close proximity. Dual-gear (two wheels) and dual-tandem-gear (four wheels) configurations are commonly used. As indicated in Figure 14, a total number of four main gear wheels are associated with two dual-gear configurations, and eight main gear wheels with two dual-tandem-gear configurations. For the case of multiple wheels, the total strain or stress in the pavement beneath one wheel is due in part to the presence of the other wheels. The maximum values of the stress and the strain at some depth in the pavement occur at a point between the wheels of the gear configuration, but these maximum values of the stress and the strain in the pavement are to a good approximation equal to the values of the stress and the strain in the pavement beneath one of the wheels of a multiple-wheel configuration. The multiple-wheel

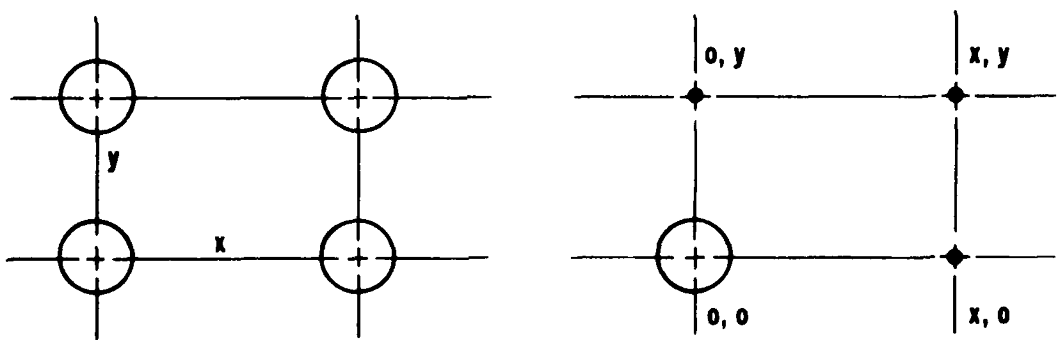
DUAL WHEELS



$\sigma_{RD} = \sigma_{RD}(0,0,h_1)$ — PCC PAVEMENT

$$\left.\begin{aligned} \epsilon_{RD} &= \epsilon_{RD}(0,0,h_1) \\ \epsilon_{VD} &= \epsilon_{VD}(0,0,H_S) \end{aligned}\right\} \text{AC PAVEMENT}$$

DUAL-TANDEM WHEELS



$\sigma_{RDT} = \sigma_{RDT}(0,0,h_1)$ — PCC PAVEMENT

$$\left.\begin{aligned} \epsilon_{RDT} &= \epsilon_{RDT}(0,0,h_1) \\ \epsilon_{VDT} &= \epsilon_{VDT}(0,0,H_S) \end{aligned}\right\} \text{AC PAVEMENT}$$

Figure 14. Calculation of the total stress and strain for dual and dual-tandem wheels

calculations are made within this approximation. The calculation of the allowable load-carrying capacity and the required overlay thickness must include the additive stress and strain effects associated with multiple-wheel loadings.

The effects of multiple-wheel loadings are accounted for by calculating the net stress and strain in the pavement or the subgrade under a selected wheel and by adding the stress and strain components of the remaining wheels occurring under the selected wheel. The BISAR computer program calculates the stress and the strain in a pavement at any depth directly under one wheel due to the action of the wheel loads applied at the pavement surface. For dual wheels, let

ε_{VD} = total vertical strain at a point in the pavement directly under one wheel at the top of the subgrade for AC pavements

ε_{RD} = total radial strain under one wheel at the bottom of the AC pavement layer

σ_{RD} = total radial stress at the bottom of the PCC layer at a point under one wheel

For dual-tandem wheels, let

ε_{VDT} = total vertical strain at the top of the subgrade for AC pavements at a point directly under one wheel

ε_{RDT} = total radial strain at the bottom of the AC pavement layer at a point under one wheel

σ_{RDT} = total radial stress at the bottom of the PCC layer at a point under one wheel

For dual wheels, the conditions that determine the allowable load-carrying capacity and the required overlay thickness are

$$\left.\begin{aligned} \varepsilon_{VD}(0,0,H_s) &= \varepsilon_{VL} \\ \varepsilon_{RD}(0,0,h_1) &= \varepsilon_{RL} \end{aligned}\right\} \quad \text{AC pavements} \tag{7}$$

$$\sigma_{RD}(0,0,h_1) = \sigma_{RL} \quad \text{PCC pavements} \tag{8}$$

while for dual-tandem wheels the conditions are

$$\left.\begin{aligned} \epsilon_{VDT}(0,0,H_s) &= \epsilon_{VL} \\ \epsilon_{RDT}(0,0,h_1) &= \epsilon_{RL} \end{aligned}\right\} \quad \text{AC pavement} \tag{9}$$

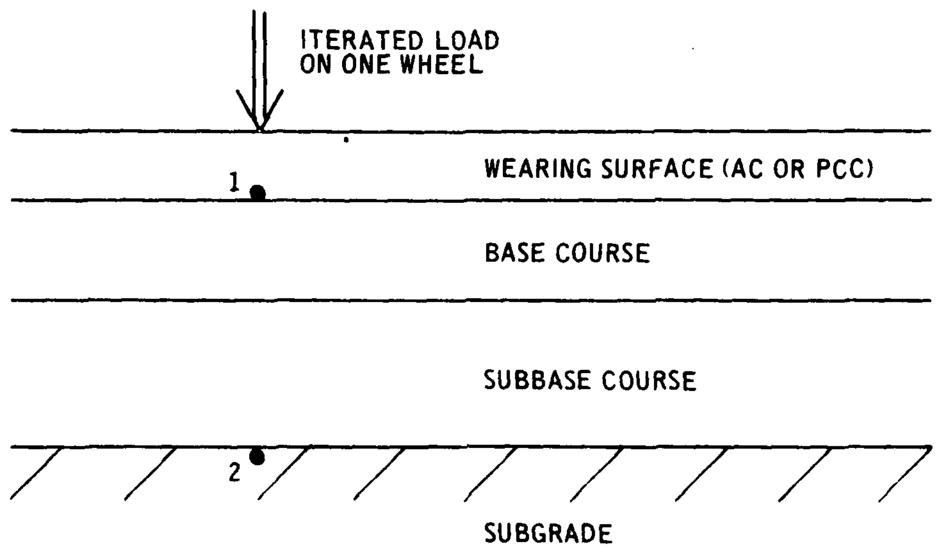
$$\sigma_{RDT}(0,0,h_1) = \sigma_{RL} \quad \text{PCC pavement} \tag{10}$$

The limiting stress and strain values do not depend on the type of surface loading and are valid for single- and multiple-wheel loadings.

ALLOWABLE LOAD-CARRYING AND REQUIRED OVERLAY THICKNESS FOR AC AND PCC PAVEMENTS

For AC and PCC pavements, the allowable load-carrying capacity is calculated by monitoring the stress and the strain at points in the pavement indicated in Figure 15. The PAVEVAL computer program has the capability of iterating the wheel load at the pavement surface for a given pavement structure until the calculated value of the vertical strain at the top of the subgrade of an AC pavement is equal to the limiting value of the vertical strain or until the calculated value of the tensile strain at the bottom of the AC layer is equal to the limiting value of the tensile strain, as shown in Figures 5, 6, and 7. The calculated value of the tensile stress at the bottom of the PCC layer must equal the limiting value of the tensile stress as given in Equation 2. This determines the allowable load-carrying capacity for AC and PCC pavements.

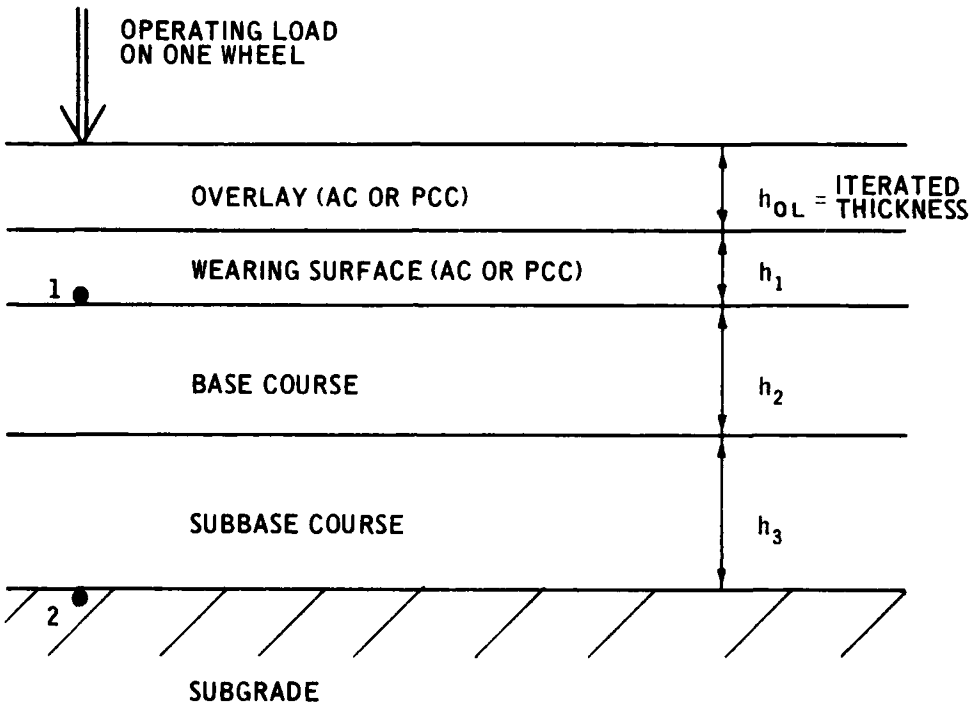
The required overlay thickness for AC and PCC pavements is calculated by examining the stress and the strain at points in the pavement indicated in Figure 16. The PAVEVAL computer program can be used to iterate the thickness of the overlay for a wheel load until the calculated value of the vertical strain at the top of the subgrade of an AC pavement is equal to or less than the limiting value of the vertical strain and until the calculated value of the tensile strain at the bottom of the AC layer is equal to or less than the limiting value of the tensile strain. For PCC pavements, the calculated value of the tensile stress at the bottom of the PCC layer must equal the limiting value of the tensile stress.



1 POINT WHERE TENSILE STRESS AND TENSILE STRAIN IS MONITORED FOR A PCC OR AN AC WEARING SURFACE

2 POINT WHERE VERTICAL COMPRESSIVE STRAIN IS MONITORED FOR AN AC WEARING SURFACE

Figure 15. Location of points where the total stress and strain are monitored for the calculation of the allowable load-carrying capacity of AC and PCC pavements



1 POINT WHERE TENSILE STRESS AND TENSILE STRAIN IS MONITORED FOR A PCC OR AN AC WEARING SURFACE

2 POINT WHERE VERTICAL COMPRESSIVE STRAIN IS MONITORED FOR AN AC WEARING SURFACE

Figure 16. Location of points where the total stress and strain are monitored for the calculation of the required overlay thickness

NUMERICAL VALUES OF THE ALLOWABLE LOAD-CARRYING CAPACITY AND THE REQUIRED OVERLAY THICKNESS

The pavement evaluation procedures discussed previously were applied to a number of PCC and AC pavement structures for single- and multiple-wheel loadings, and the allowable load-carrying capacity and the required overlay thickness were calculated by combining the layered elastic theory with inputs from vibratory nondestructive testing. For these pavement structures, the allowable load-carrying capacity and the required overlay thickness were also calculated by the conventional CBR and DSM methods for AC pavements and by the Westergaard and DSM methods for PCC pavements.

Tables 6-17 present the predicted values of the allowable load-carrying capacity and the required overlay thickness for the AC and PCC pavement sites whose structures appear in Tables 4 and 5. In these tables, the load-carrying capacity is expressed in terms of the load on one wheel. The total allowable airplane load is obtained by the expression: (allowable load on one wheel) × (total number of main gear wheels)/0.95. Single-wheel, dual-wheel, and dual-tandem-wheel gear configurations were considered. The total number of main gear wheels for these configurations are 2, 4, and 8, respectively. Specific calculations were done for the single-wheel load, the Boeing 727 (dual wheels), the DC-8-63F (dual-tandem wheels), and the DC-10-10 (dual-tandem wheels).

For AC pavements, Figures 17 and 18 compare the layered elastic theory calculation of the allowable load on one wheel and the required AC overlay thicknes with the corresponding CBR calculation of these quantities. For PCC pavements, Figures 19 and 20 compare the results of the layered elastic theory calculation of the allowable load on one wheel and the AC and PCC required overlay thickness with the results of the corresponding Westergaard calculations of these quantities. Reference 1 describes the CBR and Westergaard methods of calculating the allowable load and the overlay thickness, respectively, for AC and PCC pavements. The data plotted in Figures 17-21 correspond to the data presented in Tables 6-17.

Table 6. Allowable Load (Layered Elastic Theory Method) of AC Pavement, 1200 Annual Strain Repetitions

Site	Measured DSM kips/in.	Temperature* Adjusted DSM kips/in.	Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips	
					DC-8-63F	DC-10-10
B2	700	805	87	72	69	84
W1	860	1080	41	37	36	42
Alum Creek-11	820	541	27	25	15	28
Alum Creek-2	880	581	27	25	24	28
Alum Creek-23	1000	660	32	29	29	33
Alum Creek-32	1230	812	27	25	25	28
W2	1940	2060	79	69	67	79
P14	2120	2630	92	78	75	90
P13	2780	3000	101	85	81	98
N23	980	700	38	36	36	39
B3	1680	1200	88	76	74	88
B1	3120	2680	118	92	81	103
N18	770	630	21	20	20	23

* From Reference 1.

Table 7. Allowable Load (CBR Method) of AC Pavement, 1200 Annual Strain Repetitions

Site	Measured DSM kips/in.	Temperature* Adjusted DSM kips/in.	Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips	
					DC-8-63F	DC-10-10
B2	700	805	62	45	34	40
W1	860	1080	45	36	27	38
Alum Creek-11	820	541	10	9.7	5.8	11
Alum Creek-2	880	581	10	9.7	5.8	11
Alum Creek-23	1000	660	20	16	12	19
Alum Creek-32	1230	812	20	16	12	19
W2	1940	2060	76	59	43	59
P14	2120	2630	92	66	50	69
P13	2780	3000	106	74	57	75
N23	980	700	30	23	18	26
B3	1680	1200	65	71	53	72
B1	3120	2680	90	62	45	62
N18	770	630	19	14	10	14

* From Reference 1.

Table 8. Allowable Load (DSM Method) of AC Pavement, 1200 Annual Strain Repetitions

Site	Measured DSM kips/in.	Temperature* Adjusted DSM kips/in.	Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips	
					DC-8-63F	DC-10-10
B2	700	805	36	27	22	23
W1	860	1080	50	38	30	39
Alum Creek-11	820	541	28	21	16	21
Alum Creek-2	880	581	27	22	18	23
Alum Creek-23	1000	660	31	24	20	25
Alum Creek-32	1230	812	37	29	24	31
W2	1940	2060	91	79	55	70
P14	2120	2630	120	105	72	92
P13	2780	3000	134	108	78	99
N23	980	700	32	22	19	25
B3	1680	1200	54	66	28	36
B1	3120	2680	122	153	66	82
N18	770	630	30	23	20	25

* From Reference 1.

Table 7. Allowable Load (CBR Method) of AC Pavement, 1200 Annual Strain Repetitions

Site	Measured DSM kips/in.	Temperature* Adjusted DSM kips/in.	Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips	
					DC-8-63F	DC-10-10
B2	700	805	62	45	34	40
W1	860	1080	45	36	27	38
Alum Creek-11	820	541	10	9.7	5.8	11
Alum Creek-2	880	581	10	9.7	5.8	11
Alum Creek-23	1000	660	20	16	12	19
Alum Creek-32	1230	812	20	16	12	19
W2	1940	2060	76	59	43	59
P14	2120	2630	92	66	50	69
P13	2780	3000	106	74	57	75
N23	980	700	30	23	18	26
B3	1680	1200	65	71	53	72
B1	3120	2680	90	62	45	62
N18	770	630	19	14	10	14

* From Reference 1.

Table 9. Required Overlay Thickness (Layered Elastic Theory Method) of AC Pavement, 1200 Annual Strain Repetitions

Site	Measured DSM kips/in.	Temperature* Adjusted DSM kips/in.	Required Overlay Thickness (Single-Wheel Load) in.	Required Overlay Thickness (Dual-Wheel Load) Boeing 727 in.	Required Overlay Thickness (Dual-Tandem-Wheel Load) in.	
					DC-8-63F	DC-10-10
			SWL = 35,625 lb	SWL = 41,090 lb	SWL = 42,510 lb	SWL = 51,420 lb
B2	700	805	0.0	0.0	0.0	0.0
W1	860	1080	0.0	0.0	0.0	0.0
Alum Creek-11	820	541	2.0	4.5	5.0	5.0
Alum Creek-2	880	581	2.0	4.5	5.0	5.0
Alum Creek-23	1000	660	1.0	3.0	3.5	3.5
Alum Creek-32	1230	812	2.0	4.0	4.5	5.0
W2	1940	2060	0.0	0.0	0.0	0.0
P14	2120	2630	0.0	0.0	0.0	0.0
P13	2780	3000	0.0	0.0	0.0	0.0
N23	980	700	0.0	0.0	0.0	0.0
B3	1680	1200	0.0	0.0	0.0	0.0
B1	3120	2680	0.0	0.0	0.0	0.0
N18	770	630	5.0	6.5	7.0	7.0

* From Reference 1.

Table 10. Required Overlay Thickness (CBR Method) of AC Pavement, 1200 Annual Strain Repetitions

Site	Measured DSM kips/in.	Temperature* Adjusted DSM kips/in.	Required Overlay Thickness (Single-Wheel Load) in.	Required Overlay Thickness (Dual-Wheel Load) Boeing 727 in.	Required Overlay Thickness (Dual-Tandem-Wheel Load) in.	
					DC-8-63F	DC-10-10
			SWL = 35,625 lb	SWL = 41,090 lb	SWL = 42,510 lb	SWL = 51,420 lb
B2	700	805	0	0	2.6	1.9
W1	860	1080	0	0.78	3.2	2.9
Alum Creek-11	820	541	4.0	9.8	14	12
Alum Creek-2	880	581	4.0	9.8	14	12
Alum Creek-23	1000	660	3.1	6.1	9.3	8.2
Alum Creek-32	1230	812	3.1	6.1	9.3	8.2
W2	1940	2060	0	0	0	0
P14	2120	2630	0	0	0	0
P13	2780	3000	0	0	0	0
N23	980	700	0.2	3.5	6.4	5.5
B3	1680	1200	0	0	0	0
B1	3120	2680	0	0	0	0
N18	770	630	1.5	6.5	8.4	9.0

* From Reference 1.

Table 11. Required Overlay Thickness (DSM Method) of AC Pavement, 1200 Annual Strain Repetitions

Site	Measured DSM kips/in.	Temperature* Adjusted DSM kips/in.	Required Overlay Thickness (Single-Wheel Load) in.	Required Overlay Thickness (Dual-Wheel Load) Boeing 727 in.	Required Overlay Thickness (Dual-Tandem-Wheel Load) in.	
					DC-8-63F	DC-10-10
			SWL = 35,625 lb	SWL = 41,090 lb	SWL = 42,510 lb	SWL = 51,420 lb
B2	700	805	0	4.0	7.3	8.2
W1	860	1080	0	4.1	3.0	2.0
Alum Creek-11	820	541	2.8	5.4	7.9	6.7
Alum Creek-2	880	581	1.7	4.7	7.6	6.2
Alum Creek-23	1000	660	0.8	1.8	6.7	5.6
Alum Creek-32	1230	812	0	1.3	5.0	3.7
W2	1940	2060	0	0	0	0
P14	2120	2630	0	0	0	0
P13	2780	3000	0	0	0	0
N23	980	700	0.4	4.1	7.0	6.2
B3	1680	1200	0	0	5.4	4.8
B1	3120	2680	0	0	0	0
N18	770	630	1.0	4.4	6.9	5.5

* From Reference 1.

Table 12. Allowable Load (Layered Elastic Theory Method) of PCC Pavement, 1200 Annual Stress Repetitions

Site	Measured DSM kips/in.	R psi	Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips	
					DC-8-63F	DC-10-10
N7	1866	810	26	22	19	22
N9	1920	724	33	29	26	28
N11	1050	894	24	21	19	20
N8	1600	880	29	24	21	21
J5	1760	966	36	33	28	29
J3	2060	740	26	21	19	21
I20	2300	984	42	37	33	41
N15	2640	864	36	31	28	30
I13	3240	870	54	47	42	45
J1	2940	754	32	29	25	26
I8	3100	1003	66	60	52	56
H7	3480	929	62	56	49	51

(Continued)

Table 12 (Concluded)

Site	Measured DSM kips/in.	R psi	Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips	
					DC-8-63F	DC-10-10
H1	3520	868	67	57	53	59
H11	3700	894	67	59	53	56
H3	4360	929	59	51	47	50

Table 13. Allowable Load (Westergaard Method) of PCC Pavement, 1200 Annual Stress Repetitions

Site	Measured DSM kips/in.	R psi	Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips	
					DC-8-63F	DC-10-10
N7	1866	810	24	15	18	36
N9	1920	724	36	24	24	48
N11	1050	894	24	15	15	33
N8	1600	880	26	18	20	37
J5	1760	966	48	36	38	58
J3	2060	740	26	17	17	34
I20	2300	984	48	38	43	59
N15	2640	864	27	23	21	39
I13	3240	870	55	38	42	59
J1	2940	754	37	25	23	42
I8	3100	1003	66	50	55	69
H7	3480	929	64	48	56	71

(Continued)

Table 13 (Concluded)

Site	Measured DSM kips/in.	R psi	Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips	
					DC-8-63F	DC-10-10
H1	3520	868	81	55	58	75
H11	3700	894	67	50	62	69
H3	4360	929	74	55	62	74

Table 14. Allowable Load (DSM Method) of PCC Pavement, 1200 Annual Stress Repetitions

Site	Measured DSM kips/in.	R psi	Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips	
					DC-8-63F	DC-10-10
N7	1866	810	35	26	32	40
N9	1920	724	36	26	30	39
N11	1050	894	20	15	18	23
N8	1600	880	30	17	26	34
J5	1760	966	32	24	34	34
J3	2060	740	39	24	34	44
I20	2300	984	43	30	37	48
N15	2640	864	49	33	34	35
I13	3240	870	61	42	44	61
J1	2940	754	55	38	42	58
I8	3100	1003	57	41	44	60
H7	3480	929	65	48	56	71

(Continued)

Table 14 (Concluded)

Site	Measured DSM kips/in.	R psi	Allowable Load on One Wheel (Single Wheel) kips	Allowable Load on One Wheel (Dual Wheels) Boeing 727 kips	Allowable Load on One Wheel (Dual-Tandem Wheels) kips	
					DC-8-63F	DC-10-10
H1	3520	868	66	43	43	58
H11	3700	894	66	50	57	75
H3	4360	929	82	58	62	85

Table 15. Required Overlay Thickness (Layered Elastic Theory Method) of PCC Pavement, 1200 Annual Stress Repetition.

Site	Measured DSM kips/in.	k psi	Overlay Material	Required Overlay Thickness (Single-Wheel Load) in. SWL = 35,625 lb	Required Overlay Thickness (Dual-Wheel Load) Boeing 727 in. SWL = 41,[illegible]90 lb	Required Overlay Thickness ([illegible]-Wheel Load) in. [illegible]-3-63F SWL = [illegible]510 lb	Required Overlay Thickness ([illegible]-Wheel Load) in. DC-10-10 SWL = 51,420 lb
N7	1866	810	AC	6.0	8.5	[illegible]	13.0
			PCC	2.5	4.0	5.5	6.5
N9	1920	724	AC	2.5	7.5	9.0	10.0
			PCC	1.0	3.5	4.0	4.5
N11	1050	894	AC	6.0	9.0	11.0	13.0
			PCC	2.5	4.5	5.5	6.0
N8	1600	880	AC	4.0	7.5	9.0	12.0
			PCC	1.5	3.5	4.5	6.0
J5	1760	966	AC	0.0	4.5	7.0	9.0
			PCC	0.0	2.0	3.0	4.0
J3	2060	740	AC	5.5	8.5	11.0	13.0
			PCC	2.5	4.0	6.0	7.0
I20	2300	984	AC	0.0	X	X	X
			PCC	0.0			
N15	2640	864	AC	0.0	5.5	7.5	8.5
			PCC	0.0	2.5	3.0	3.5

(Continued)

Table 15 (Concluded)

Site	Measured DSM kips/in.	R psi	Overlay Material	Required Overlay Thickness (Single-Wheel Load) in.	Required Overlay Thickness (Dual-Wheel Load) Boeing 727 in.	Required Overlay Thickness (Dual-Tandem-Wheel Load) in.	
				SWL = 35,625 lb	SWL = 41,090 lb	DC-8-63F SWL = 42,510 lb	DC-10-10 SWL = 51,420 lb
I13	3240	870	AC	0.0	0.0	0.0	0.0
			PCC	0.0	0.0	0.0	0.0
J1	2940	754	AC	2.5	7.0	9.0	11.0
			PCC	1.0	3.0	4.0	5.0
I8	3100	1003	AC	0.0	0.0	0.0	0.0
			PCC	0.0	0.0	0.0	0.0
H7	3480	929	AC	0.0	0.0	0.0	0.0
			PCC	0.0	0.0	0.0	0.0
H1	3520	868	AC	0.0	0.0	0.0	0.0
			PCC	0.0	0.0	0.0	0.0
H11	3700	894	AC	0.0	0.0	0.0	0.0
			PCC	0.0	0.0	0.0	0.0
H3	4360	929	AC	0.0	0.0	0.0	0.0
			PCC	0.0	0.0	0.0	0.0

Table 16. Required Overlay Thickness (Westergaard Method) of PCC Pavement, 1200 Annual Stress Repetitions

Site	Measured DSM kips/in.	R psi	Overlay Material	Required Overlay Thickness (Single-Wheel Load) in.	Required Overlay Thickness (Dual-Wheel Load) Boeing 727 in.	Required Overlay Thickness (Dual-Tandem-Wheel Load) in.	
						DC-8-63F	DC-10-10
				SWL = 35,625 lb	SWL = 41,090 lb	SWL = 42,510 lb	SWL = 51,420 lb
N7	1866	810	AC	3.5	13	11	7.4
			PCC	2.9	8.0	7.1	4.9
N9	1920	724	AC	0	8.4	7.8	1.5
			PCC	0	6.1	5.7	1.7
N11	1050	894	AC	4.0	12	13	8.4
			PCC	3.2	7.1	7.8	5.6
N8	1600	880	AC	2.5	10	8.5	6.0
			PCC	2.3	6.4	5.6	4.3
J5	1760	966	AC	3.3	2.0	2.0	0
			PCC	2.8	2.1	2.1	0
J3	2060	740	AC	3.3	12	14	8.4
			PCC	2.8	7.8	8.7	5.8
I20	2300	984	AC	0	2.2	0	0
			PCC	0	2.5	0	0
N15	2640	864	AC	0	8.6	11	6.1
			PCC	0	6.2	7.6	4.8

(Continued)

Table 16 (Concluded)

Site	Measured DSM kips/in.	R psi	Overlay Material	Required Overlay Thickness (Single-Wheel Load) in.	Required Overlay Thickness (Dual-Wheel Load) Boeing 727 in.	Required Overlay Thickness (Dual-Tandem-Wheel Load) in.	
						DC-8-63F	DC-10-10
				SWL = 35,625 lb	SWL = 41,090 lb	SWL = 42,510 lb	SWL = 51,420 lb
I13	3240	870	AC	0	1.2	0	0
			PCC	0	2.0	0	0
J1	2940	754	AC	3.3	8.5	10	3.9
			PCC	2.8	6.2	7.1	3.6
I8	3100	1003	AC	0	0	0	0
			PCC	0	0	0	0
H7	3480	929	AC	0	0	0	0
			PCC	0	0	0	0
H1	3520	868	AC	0	0	0	0
			PCC	0	0	0	0
H11	3700	894	AC	0	0	0	0
			PCC	0	0	0	0
H3	4360	929	AC	0	0	0	0
			PCC	0	0	0	0

Table 17. Required Overlay Thickness (DSM Method) of PCC Pavement, 1200 Annual Stress Repetitions

Site	Measured DSM kips/in.	R psi	Overlay Material	Required Overlay Thickness (Single-Wheel Load) in.	Required Overlay Thickness (Dual-Wheel Load) Boeing 727 in.	Required Overlay Thickness (Dual-Tandem-Wheel Load) in.	
						DC-8-63F	DC-10-10
				SWL = 35,625 lb	SWL = 41,090 lb	SWL = 42,510 lb	SWL = 51,420 lb
N7	1866	810	AC	1.2	9.0	7.5	3.8
			PCC	1.4	5.9	5.1	3.1
N9	1920	724	AC	0	12.5	12.0	6.8
			PCC	0	8.2	8.0	5.2
N11	1050	894	AC	7.8	18	18	12
			PCC	5.3	10	10	7.6
N8	1600	880	AC	2.8	12	9.5	6.5
			PCC	2.5	7.4	6.2	4.6
J5	1760	966	AC	3.2	16	18.5	12
			PCC	3.0	10	11	8.1
J3	2060	740	AC	0	8.8	7.5	3.5
			PCC	0	6.0	5.3	3.0
I20	2300	984	AC	0	8.0	7.5	3.0
			PCC	0	5.9	5.6	2.8
N15	2640	864	AC	0	5.8	5.8	1.8
			PCC	0	4.6	4.6	2.0

(Continued)

Table 17 (Concluded)

Site	Measured DSM kips/in.	R psi	Overlay Material	Required Overlay Thickness (Single-Wheel Load) in.	Required Overlay Thickness (Dual-Wheel Load) Boeing 727 in.	Required Overlay Thickness (Dual-Tandem-Wheel Load) in.	
						DC-8-63F	DC-10-10
				SWL = 35,625 lb	SWL = 41,090 lb	SWL = 42,510 lb	SWL = 51,420 lb
I13	3240	870	AC	0	0	0	0
			PCC	0	0	0	0
J1	2940	754	AC	0	2.2	0	0
			PCC	0	2.3	0	0
I8	3100	1003	AC	0	0	0	0
			PCC	0	0	0	0
H7	3480	929	AC	0	0	0	0
			PCC	0	0	0	0
H1	3520	868	AC	0	0	0	0
			PCC	0	0	0	0
H11	3700	894	AC	0	0	0	0
			PCC	0	0	0	0
H3	4360	929	AC	0	0	0	0
			PCC	0	0	0	0

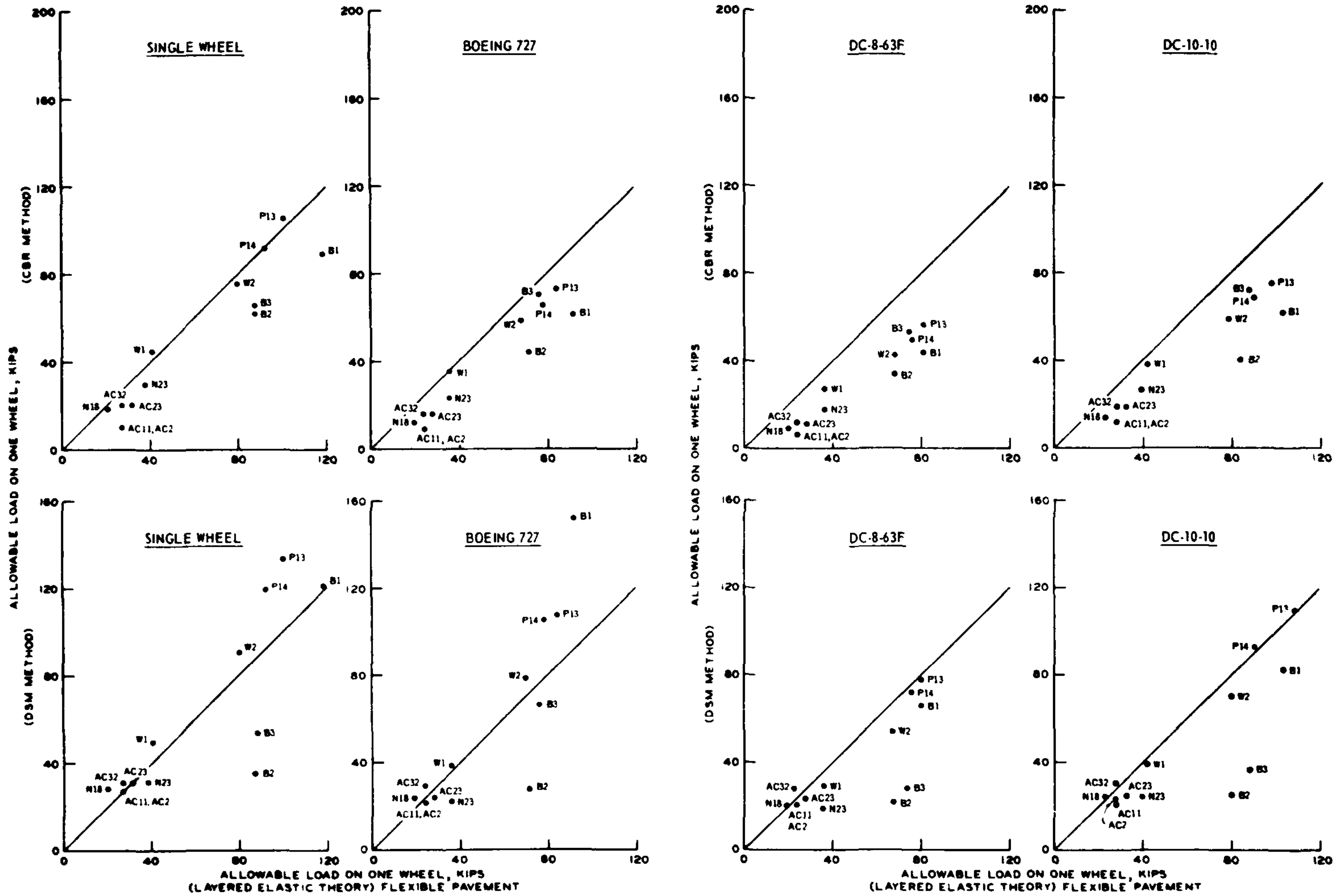


Figure 17. Allowable load-carrying capacity of AC pavements calculated by the layered elastic theory and by the CBR and DSM methods

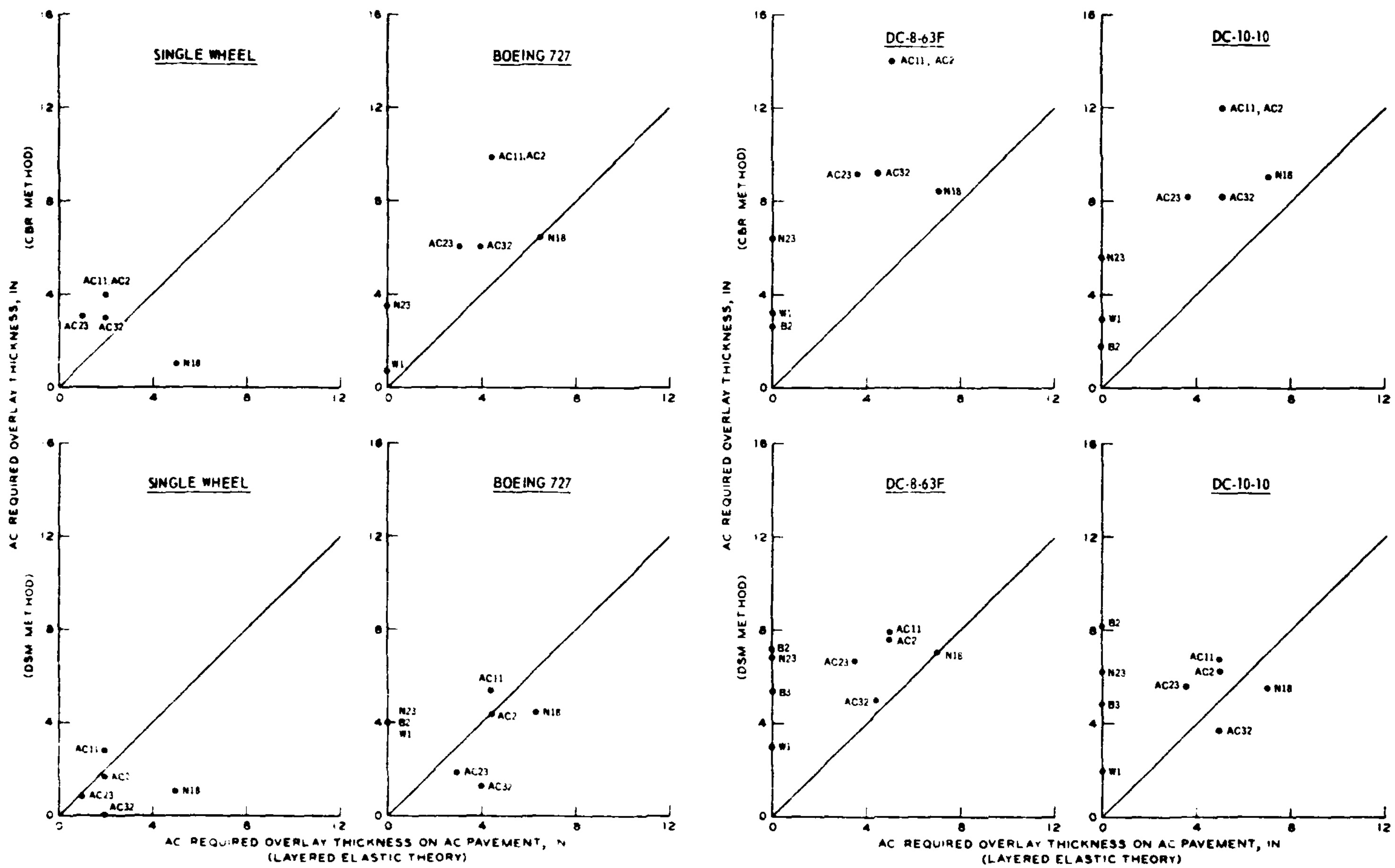


Figure 18. AC required overlay thickness for AC pavements calculated by the layered elastic theory and by the CBR and DSM methods

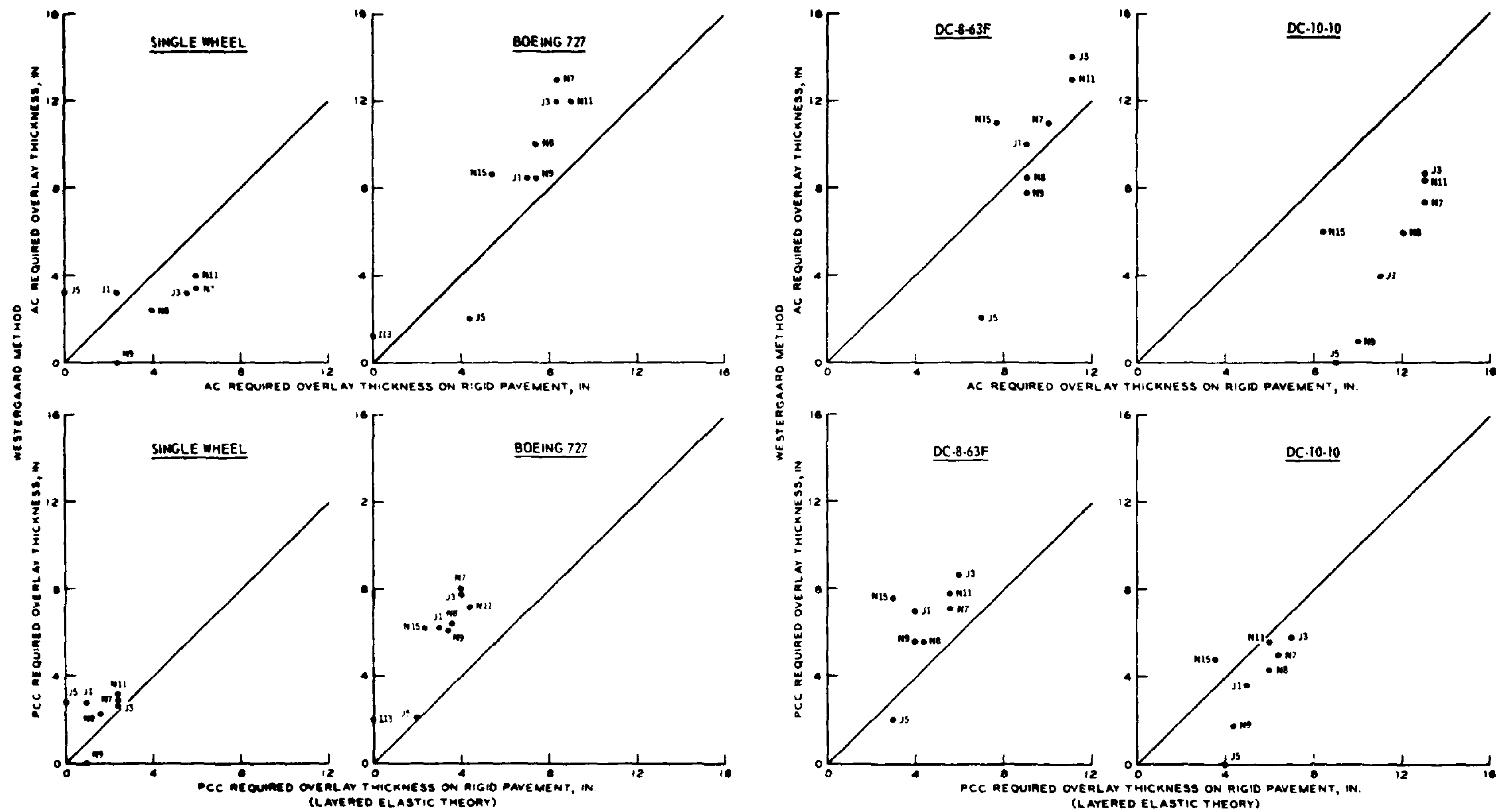


Figure 19. AC and PCC required overlay thicknesses for PCC pavements calculated by the layered elastic theory and the Westergaard method

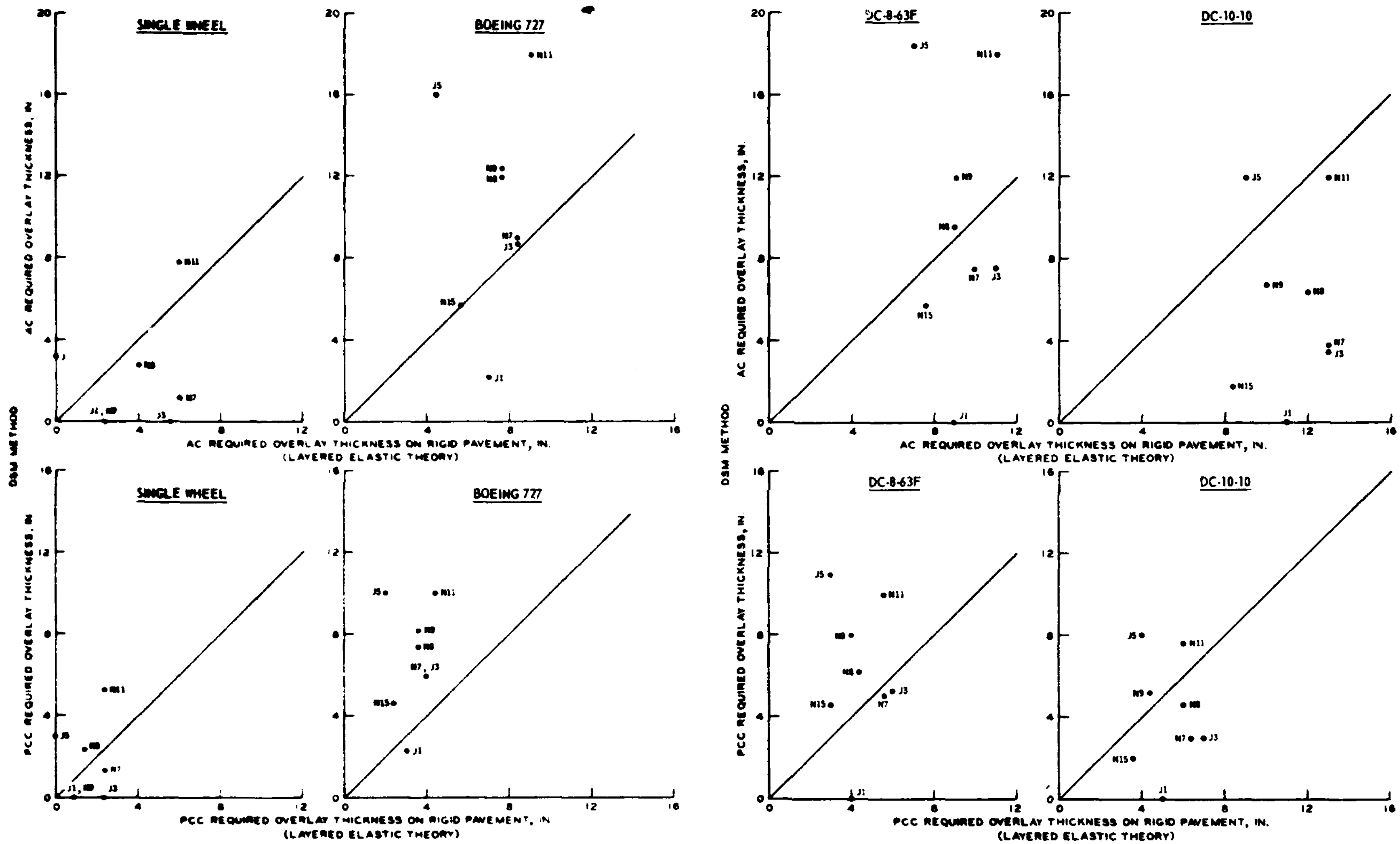


Figure 20. AC and PCC required overlay thicknesses for PCC pavements calculated by the layered elastic theory and the DSM method

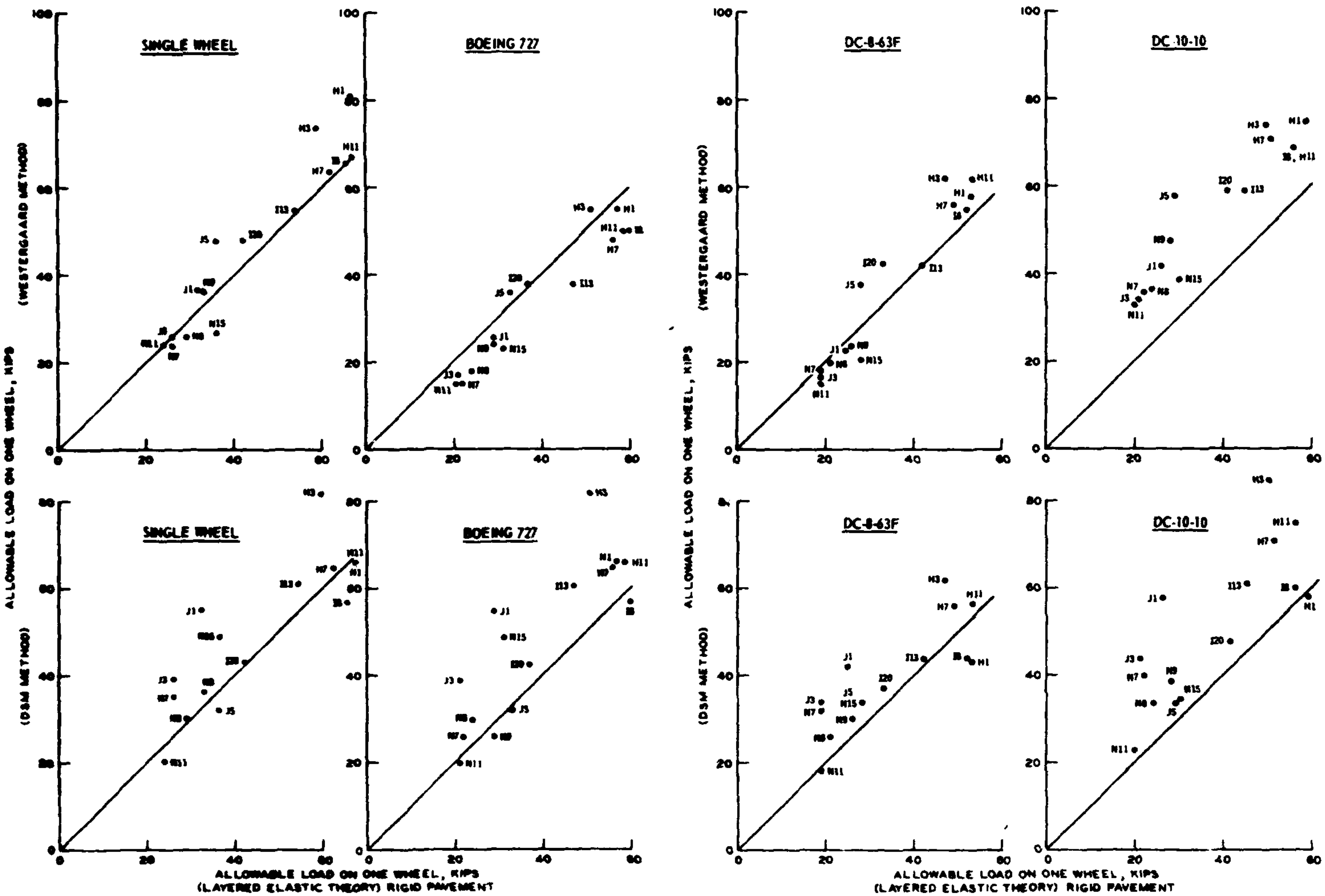


Figure 21. Allowable load-carrying capacity of PCC pavements calculated by the layered elastic theory and by the DSM and Westergaard methods

For some AC pavements, the layered elastic theory method predicts values of allowable loads that are larger than the aircraft loads, while the DSM method predicts allowable load values that are less than the aircraft load (Tables 6 and 8 and Figure 17). For these cases, the values of the required overlay thicknessess predicted by the layered elastic theory method are zero while those predicted by the DSM method have nonzero values (Figure 18).

SUMMARY, CONCLUSIONS, AND RECOMMENDATIONS

SUMMARY

The capability of determining the load-carrying capacity of a pavement and the overlay thickness required to upgrade a pavement is important to pavement engineers. A simple method of pavement evaluation combining vibratory nondestructive field tests with a layered elastic theory was developed to satisfy the needs of the pavement engineer. The layered elastic theory approach to calculating the required overlay thickness and the load-carrying capacity of a pavement requires the value of the subgrade Young's modulus, and this value is determined by an analysis of the pavement dynamic response obtained from vibratory nondestructive testing. This approach requires a knowledge of the structure of the pavement and the subgrade as described by the elastic modulus, the Poisson's ratio, and the thickness of each pavement layer.

The method of pavement evaluation presented consists of two parts: (a) the determination of the subgrade Young's modulus from vibratory nondestructive tests that measure the pavement response to an applied dynamic load, and (b) the use of the layered elastic theory and the determined value of the subgrade Young's modulus to predict the allowable load-carrying capacity and the required overlay thickness of a pavement. Two computer programs, SUBE and PAVEVAL, are used to evaluate a pavement based on vibratory nondestructive tests and the layered elastic theory.

The computer program SUBE determines the value of the subgrade Young's modulus from the measured dynamic load-deflection curves and the estimated values of the elastic moduli and thicknesses of the pavement layers. The mathematical model on which SUBE is based is a nonlinear harmonic oscillator whose predicted dynamic load-deflection curve is matched to the measured dynamic load-deflection curves in order to determine the value of the subgrade Young's modulus. The predicted values of the subgrade Young's modulus are in essential agreement with the formula $E_s = 1500$ CBR and are not especially sensitive to the choice of the elastic moduli of the pavement layers.

The computer program PAVEVAL calculates the allowable load-carrying capacity and the required overlay thickness values for the layered elastic theory by relating the stress and the strain at any point in the pavement or subgrade to the magnitude of the static load applied to the pavement surface. The elastic moduli, Poisson's ratios, and thicknesses of the pavement layers and the subgrade must be known to use this computer program. For PCC, the flexural strength must also be known. Aircraft parameters including the load on one wheel, the tire contact area, wheel spacings, and the total number of main gear wheels are also required for PAVEVAL.

CONCLUSIONS

The study of predicting pavement performance and overlay design by the combined techniques of layered elastic theory and vibratory nondestructive testing yielded the following conclusions:

a. The layered elastic theory method using the subgrade Young's modulus determined from the results of vibratory nondestructive tests is sufficient to predict the allowable load-carrying capacity and the required overlay thickness for a pavement; computer programs SUBE and PAVEVAL have been developed to aid in pavement evaluation and overlay design.

b. The value of the subgrade Young's modulus can be obtained from vibratory nondestructive test results through the use of the computer program SUBE.

c. Limiting stress and strain criteria can be used in conjunction with the layered elastic theory to determine the allowable load-carrying capacity and the required overlay thickness of a pavement. This can be determined for dual-wheel and dual-tandem-wheel loads, as well as for single-wheel loads, by using the computer program PAVEVAL.

RECOMMENDATIONS

A method has been developed for calculating the allowable load-carrying capacity and the required overlay thickness for pavements by using the combined methods of layered elastic theory and vibratory nondestructive testing. The accuracy of these calculations depends in part on the accuracy of the predicted values of the subgrade Young's

modulus. Further experimental work is necessary to validate the predicted pavement evaluations, overlay designs, and values of the subgrade Young's modulus.

DETERMINATION OF SUBSURFACE STRUCTURE

The determination of the subgrade Young's modulus by the vibratory nondestructive testing technique requires a knowledge of the elastic moduli of the pavement layers above the subgrade. The determination of the allowable load-carrying capacity of a pavement by the layered elastic theory method requires the elastic moduli of all pavement layers as well as the Young's modulus of the subgrade. Therefore, the Young's moduli of the pavement layers are used twice in the procedure for calculating the allowable load-carrying capacity of a pavement. In view of this, it is recommended that:

a. Vibratory nondestructive tests be developed that will accurately determine the values of the Young's moduli of all pavement layers.

b. A reliable method be developed to estimate the Young's modulus of the material in each pavement layer in terms of its composition and structure.

STATIC LOAD TESTS

Static load tests are required for the conventional evaluation of PCC and AC pavements. These tests determine the CBR for the AC pavement evaluation and the coefficient of the subgrade reaction for the PCC evaluation using the Westergaard theory. It is recommended that static load tests and vibratory nondestructive tests be performed at a number of pavement sites so that further comparisons can be made.

LABORATORY CONFIRMATION OF FIELD TEST DATA

A complete connection between the resilient modulus laboratory tests and the vibratory nondestructive field tests has not yet been accomplished. However, the results of a preliminary theoretical study show that it is possible to apply a nonlinear dynamic theory to the resilient modulus laboratory test to determine the static elastic

Young's modulus of a subgrade soil and to compare this value with the Young's modulus value predicted by the nonlinear dynamic analysis of the vibratory nondestructive field test data and with the Young's modulus predicted by the formula $E_s = 1500$ CBR.[4] It is recommended that resilient modulus tests be conducted on undisturbed soil samples taken at sites where vibratory nondestructive tests have been conducted.

REFERENCES

1. Green, J. L. and Hall, J. W., Jr., "Nondestructive Vibratory Testing of Airport Pavements; Evaluation Methodology and Experimental Test Results," Vol 1, Report No. FAA-RD-73-305-I, Department of Transportation, Federal Aviation Administration, Washington, D. C., 1975.

2. Green, J. L., "Literature Review - Elastic Constants for Airport Pavement Materials," Report No. FAA-RD-76-138, Department of Transportation, Federal Aviation Administration, Washington, D. C., 1978.

3. Weiss, R. A., "Nondestructive Vibratory Testing of Airport Pavements; Theoretical Study of the Dynamic Stiffness and Its Application to the Vibratory Nondestructive Method of Testing Pavements," Vol 2, Report No. FAA-RD-73-2-5-II, Department of Transportation, Federal Aviation Administration, Washington, D. C., 1975.

4. ____________, "Subgrade Elastic Moduli Determined from Vibratory Testing of Pavements," Report No. FAA-RD-76-158, Department of Transportation, Federal Aviation Administration, Washington, D. C., 1977.

5. Tomita, H., "Field NDE of Airport Pavements; Materials Evaluation," Vol XXXIII, No. 7, Department of Transportation, Federal Aviation Administration, Washington, D. C., 1975.

6. Department of Transportation, Federal Aviation Administration, "Use of Nondestructive Testing Devices in the Evaluation of Airport Pavements," Advisory Circular, AC No. 150/5370-11, Washington, D. C., 1976.

7. ____________, "Airport Pavement Design and Evaluation," Advisory Circular, AC No. 150/5320-6B, Washington, D. C., 1974.

8. Hutchinson, R. L., "Basis of Rigid Pavement Design for Military Airfields," Miscellaneous Paper No. 5-7, Department of the Army, Ohio River Division Laboratories, Corps of Engineers, Cincinnati, Ohio, 1966.

9. Barker, W. R. and Brabston, W. N., "Development of a Structural Design Procedure for Flexible Airport Pavements," Technical Report S-75-17, U. S. Army Engineer Waterways Experiment Station, CE, Vicksburg, Miss., 1975.

10. Lysmer, J., Monismith, C. L., and Watkins, D. J., "Nondestructive Pavement Evaluation by the Wave Propagation Method," Report No. TE-74-2, University of California, Berkeley, Calif., 1974.

11. Vidale, R. F., "The Dispersion of Stress Waves in Layered Media Overlying a Half Space of Lesser Acoustic Rigidity," Ph. D. Thesis, University of Wisconsin, 1964.

12. Parker, F., Barker, W. R., Gunkel, R. C., and Odom, E. C., "Development of a Structural Design Procedure for Rigid Airport Pavements," Report No. FAA-RD-77-81, Department of Transportation, Federal Aviation Administration, Washington, D. C., 1979.

13. Allen, J. A. and Thompson, M. R., "The Effects of Non-Constant Lateral Pressures on the Resilient Response of Granular Materials," Department of Civil Engineering, University of Illinois, Urbana, Ill., 1973.

14. Thompson, M. R. and Robnett, Q. L., "Resilient Properties of Subgrade Soils," Final Report No. UILU-ENG-76-2009, Transportation Research Laboratory, Department of Civil Engineering, University of Illinois, Urbana, Ill., 1976.

15. ____________, "Data Summary Resilient Properties of Subgrade Soils," Final Report UILU-ENG-76-2009, Transportation Research Laboratory, Department of Civil Engineering, University of Illinois, Urbana, Ill., 1976.

16. Brown, D. N. and Thompson, O. O., "Lateral Distribution of Aircraft Traffic," Miscellaneous Paper S-73-56, U. S. Army Engineer Waterways Experiment Station, CE, Vicksburg, Miss., 1973.

17. Heukelom, W. and Foster, C. R., "Dynamic Testing of Pavements," <u>Transactions, American Society of Civil Engineers</u>, Vol 127, Part I, 1962, pp. 425-257.

APPENDIX A: COMPUTER PROGRAM SUBE

The evaluation of rigid and flexible pavements by the combined methods of vibratory nondestructive testing and layered elastic theory requires two computer programs. The computer program SUBE calculates the value of the subgrade Young's modulus from the dynamic load-deflection curves measured at the pavement surface and from the chosen values of the elastic moduli of the pavement layers.

DOCUMENTATION OF THE COMPUTER PROGRAM SUBE

PROGRAM IDENTIFICATION

a. Program Title. WES Nonlinear Dynamic Load-Deflection Program

b. Program Code Name. SUBE

c. Writer. Richard A. Weiss and Adrian P. Park

d. Organization. U. S. Army Engineer Waterways Experiment Station, Vicksburg, MS 39180

e. Date. July 1977

f. Source Language. Fortran IV

g. Abstract. Program calculates the value of the subgrade Young's modulus by requiring a nonlinear dynamic pavement response model to agree with the measured dynamic load-deflection curves.

ENGINEERING DOCUMENTATION

Narrative Description. The pavement and subgrade are modeled as a nonlinear harmonic oscillator with third-order and fifth-order nonlinear terms. The inertial and damping characteristics of the pavement are introduced by an effective pavement mass and a damping constant. These model parameters are expressed in terms of the measured DSM of the pavement. The elastic characteristics of the pavement and subgrade are represented by the Young's moduli and Poisson's ratios of the pavement layers.

Method of Solution. The solution of a nonlinear harmonic oscillator model of pavement response is determined in terms of the elastic moduli and thicknesses of the pavement layers, the effective

mass of the pavement, the damping constant, and the assorted nonlinear model parameters. The value of the subgrade Young's modulus is obtained by matching the theoretical solution for the dynamic load-deflection curve with the measured value of the dynamic load-deflection curve. The computer program SUBE is used to calculate the subgrade Young's modulus. This computer program iterates the value of the subgrade Young's modulus until the theoretically predicted dynamic load-deflection curve agrees with the measured dynamic load-deflection curve; the value of the subgrade Young's modulus that brings agreement between its measured and theoretical load-deflection curves is the subgrade Young's modulus value that is printed out by the program SUBE.

Program Capabilities. The program calculates the subgrade Young's modulus from dynamic load-deflection curves measured on either flexible or rigid pavements. Rigid and flexible pavements can be handled by entering the appropriate values of the elastic moduli of the wearing surface. The computer program SUBE is valid only for a limited range of measured dynamic load-deflection curves. The DSM is the slope of the measured dynamic load-deflection curve for a dynamic load F_D = 15 kips. The computer program SUBE gives valid predictions of the subgrade Young's modulus only within the range $300 < DSM < 6500$ kips/in.

Printed Output. The printed output consists of the predicted value of the subgrade Young's modulus.

Computer Equipment. The program SUBE was developed on the IBM 360/65 computer.

INPUT GUIDE FOR COMPUTER PROGRAM SUBE

The input for the computer program SUBE is the elastic moduli and thicknesses of the pavement layers, the measured DSM, and a point-by-point description of the measured dynamic load-deflection curve. The point-by-point description of the measured dynamic load-deflection curve is entered into SUBE by means of a data file. The

elastic constants, layer thicknesses, and the measured DSM value are entered into the main body of the computer program. This is done as follows:

2360	Measured DSM value (kips/in.)
2365	Enter 0.0 if dynamic load-deflection curve is straight and 1.0 if dynamic load-deflection curve is curved
5240	Poisson's ratio of layer 1
5250	Poisson's ratio of layer 2
5260	Poisson's ratio of layer 3
5270	Poisson's ratio of layer 4
5300	Young's modulus of layer 1, psi
5310	Young's modulus of layer 2, psi
5320	Young's modulus of layer 3, psi
5330	Initial value of Young's modulus of subgrade, psi
5340	Iteration statement for Young's modulus of subgrade
5580	Thickness of layer 1, in.
5590	Thickness of layer 2, in.
5600	Thickness of layer 3, in.

INPUT GUIDE FOR DATA
FILE FOR SUBE

010	9000	
	Dynamic Load, kips	Dynamic Deflection, mils
012	2	"
014	4	"
016	6	"
018	8	"
020	10	"
022	12	"
024	14	"

SAMPLE PROBLEM USING
PROGRAM SUBE

The calculation of the subgrade Young's modulus using the program SUBE proceeds as follows. From the measured dynamic load-deflection curve, create the following data file and computer input for SUBE.

Date File N23		
010	9000	
012	2	1.2
014	4	2.5
016	6	3.85
018	8	5.5
020	10	7.5
022	12	9.4
024	14	11.5
SAVE N23		

Computer Input	
2360	DSMM = 980
2365	CUR = 1.0
5240	POIS1 = 0.3
5250	POIS2 = 0.3
5260	POIS3 = 0.35
5270	POIS4 = 0.35
5300	EMD1 = 1.3 x 10^6
5310	EMD2 = 1.3 x 10^6
5320	EMD3 = 4.0 x 10^4
5330	EMD4 = 10,000
5340	633 EMD4 = EMD4 + 1000
5580	H1 = 3.0
5590	H2 = 3.0
5600	H3 = 7.0
RUN	

NAME OF DATA FILE? N23

PROGRAM LISTING

A complete listing of the computer program is presented on the following pages.

```
/ JOB SUBE,ROKPDH,ROKPDH,OPT=(C,R,T)
/ LIMIT BAND=100,MIN=60
/ FTNLX SPACE=90000,FADDMEM=40K,FTNTIME=100000,FTNOPT=(K,M,X)
C  PROGRAM TITLE            WES NONLINEAR DYNAMIC LOAD-DEFLECTION    00000015
C                  CURVE PROGRAM                                     00000016
C  PROGRAM CODE NAME        SUBE                                     00000020
C  WRITER         RICHARD A. WEISS AND ARDEN P. PARK                 00000025
C  ORGANIZATION             U.S. ARMY ENGINEER WATERWAYS EXPERIMENT STATIONO
C                 VICKSBURG,MISSISSIPPI 39180                        00000035
C  DATE           OCTOBER 1978                                       00000040
C  SOURCE LANGUAGE          FORTRAN IV                               00000045
C  AVAILABILITY             COMPLETE PROGRAM LISTING IS AVAILABLE AT WES0050
C                                                                    00000055
C  ABSTRACT       COMPUTER PROGRAM SUBE CALCULATES THE VALUE OF THE SUBGRADE000060
C            YOUNG'S MODULUS FROM THE DYNAMIC  LOAD-DEFLECTION CURVES    00000065
C            MEASURED AT THE PAVEMENT SURFACE. THIS IS DONE BY       00000070
C            REQUIRING THE THEORETICAL RESULTS OF A NONLINEAR DYNAMIC    00000075
C            RESPONSE MODEL FOR THE PAVEMENT TO AGREE WITH THE       00000080
C            MEASURED DYNAMIC LOAD-DEFLECTION CURVES                 00000085
C                                                                    00000090
C          PROGRAM SUBE                                              00000095
*  POLFT DEC 6, 1972                                                 00000100
C                                                                    00000105
C  THE SUBPROGRAM POLFT FITS LEAST-SQUARES POLYNOMIALS TO            00000110
C  BIVARIATE DATA. IT IS APPLIED TO THE MEASURED DYNAMIC LOAD-       00000115
C  DEFLECTION CURVES IN THE FORM OF AN ODD ORDER POLYNOMIAL          00000120
C  HAVING LINEAR,CUBIC AND FIFTH ORDER TERMS                         00000125
C                                                                    00000130
      DIMENSION A(15),B(15),S(15),G(15),U(15),                       00000135
     & ZP(6),RR(5),CR(5),                                            00000140
     &  P(100),X(100),Y(100),C(100),Q(100)                           00000145
      INTEGER FNAME                                                  00000155
C                                                                    00000170
      FNAME=05                                                       00000175
   10 FORMAT(A3)                                                     00000180
   30 GO TO 590                                                      00000185
   36 Z=0                                                            00000190
      O=1                                                            00000195
      K=12                                                           00000200
      N=N+1                                                          00000205
      IF(N.GT.12) GO TO 576                                          00000210
      IF(M.LT.N) GO TO 616                                           00000215
      IF(M.GT.100) GO TO 570                                         00000220
      T7=Z
      T8=Z
      W7=Z
  305 DO 310 I=1,M                                                   00000230
      W7=W7+X(I)                                                     00000235
      T7=T7+Y(I)                                                     00000240
  310 T8=T8+Y(I)*Y(I)                                                00000245
      T9=(M*T8-T7*T7)/(M*M-M)                                        00000250
```

```
      W77=W7/M                                                          00000255
      T77=T7/M                                                          00000260
      T99=SQRT(T9)                                                      00000265
  314 FORMAT(///,' L E A S T   S Q U A R E S   P O L Y N O M I A L S',/,
     &//,7X,'NUMBER OF POINTS =',I2,/,7X,'MEAN VALUE OF X  =',1PE14.6,
     &/,7X,'MEAN VALUE OF Y  =',1PE14.6,/,7X,'STD ERROR OF  Y  =',
     &1PE14.6)
C                                                                       00000295
      DO 352 I=1,M                                                      00000300
      P(I)=Z                                                            00000305
  352 Q(I)=O                                                            00000310
      DO 362 I=1,11                                                     00000315
  362 A(I)=Z
      B(I)=Z
      S(I)=Z
      E1=Z
      F1=Z
      W1=M                                                              00000330
      N4=K                                                              00000335
      I=1                                                               00000340
      K1=2                                                              00000345
      IF(N.NE.0) K1=N4                                                  00000350
  380 W=Z                                                               00000355
      DO 386 L=1,M                                                      00000360
  386 W=W+Y(L)*Q(L)                                                     00000365
      S(I)=W/W1                                                         00000370
      IF(I-N4.GE.0.OR.I-M.GE.0) GO TO 428                               00000375
      E1=Z                                                              00000380
      DO 398 L=1,M                                                      00000385
  398 E1=E1+X(L)*Q(L)*Q(L)                                              00000390
      E1=E1/W1                                                          00000395
      A(I+1)=E1                                                         00000400
      W=Z                                                               00000405
      DO 416 L=1,M                                                      00000410
      V=(X(L)-E1)*Q(L)-F1*P(L)                                          00000415
      P(L)=Q(L)                                                         00000420
      Q(L)=V                                                            00000425
  416 W=W+V*V                                                           00000430
      F1=W/W1                                                           00000435
      B(I+2)=F1                                                         00000440
      W1=W                                                              00000445
      I=I+1                                                             00000450
      GO TO 380                                                         00000455
  428 DO 432 L=1,12,1                                                   00000460
  432 G(L)=Z                                                            00000465
      G(1)=O                                                            00000470
      DO 464 J=1,N                                                      00000475
      S1=Z                                                              00000480
      DO 448 L=1,N                                                      00000485
      IF(L.NE.1)G(L)=G(L)-A(L)*G(L-1)                                   00000490
      IF(L.GT.2)G(L)=G(L)-B(L)*G(L-2)                                   00000495
  448 S1=S1+S(L)*G(L)                                                   00000500
      U(J)=S1                                                           00000505
```

```
      L=N
      DO 460 I2=2,N
      G(L)=G(L-1)
      L=L-1
  460 CONTINUE
      G(1)=Z
  464 CONTINUE
      T=Z
      DO 488 L=1,M
      C(L)=Z
      J=N
      DO 482 I2=1,N
      C(L)=C(L)*X(L)+U(J)
      J=J-1
  482 CONTINUE
      T3=Y(L)-C(L)
      T=T+T3*T3
  488 CONTINUE
      IF(M.NE.N) GO TO 496
      T5=0
      GO TO 498
  496 T5=T/(M-N)
  498 Q7=1-T/(T9*(M-1))
      IF(M.NE.99)GOTO789
      ITEMP=N-1
      PRINT 500,ITEMP,Q7
  500 FORMAT(/,' POLYFIT OF DEGREE ',I2,'  INDEX OF DETERM = ',
     &1PE14.6)
      PRINT 516
  516 FORMAT(//,' TERM',8X,'COEFFICIENT',/)
      DO 526 J=1,N
      I2=J-1
  526 PRINT 527,I2,U(J)
  527 FORMAT(I4,7X,1PE14.7)
      PRINT 530
  530 FORMAT(//,' X-ACTUAL',12X,'Y-ACTUAL',3X,'Y-CALC',8X,'DIFF',
     &9X,'PCT-DIFF',/)
      DO 550 L=1,M
      Q8=Y(L)-C(L)
      IF(C(L)-0.0) 540,548,540
  540 Q88=100.0*Q8/C(L)
      PRINT 551,X(L),Y(L),C(L),Q8,Q88
      GO TO 550
  548 PRINT 552,X(L),Y(L),C(L),Q8
  550 CONTINUE
  551 FORMAT(1P5E14.6)
  552 FORMAT(1P4E14.6,3X,'INFINITE')
  555 CONTINUE
      T55=SQRT(T5)
      PRINT 553,T55
  553 FORMAT(/,10X,'STD ERROR OF ESTIMATE FOR Y = ',1PE14.6)
C
C  THIS IS THE END OF THE PART POLFT
```

```
      C
      C  TO ELIMINATE THE DIFFICULTIES ASSOCIATED WITH THE USE OF
      C  LARGE NUMBERS THE INTERNAL CALCULATIONS OF THIS PROGRAM
      C  ARE DONE IN THE UNITS LISTED AS FOLLOWS, BUT THE UNITS
      C  OF THE INPUT DATA ARE IN INCHES
      C
      C     UNITS OF MASS - KIP-SEC**2/MILL
      C     UNITS OF WEIGHT - KIP
      C     UNITS OF DAMPING - KIP-SEC/MILL
      C     UNITS OF DYNAMIC STIFFNESS - KIP/MILL
      C     UNITS OF NONLINEAR COEFF. B - KIP/MILL**3
      C     UNIT OF NONLINEAR COEFF. E -KIP/MILL**5
      C
        789 CONTINUE
            AG=384000.0
            DSMM = 980
            FDYN=15.0
            CUR = 1.0
            CAS=DSMM/50.0
            CDS=DSMM/250.0
            CZS=DSMM/500.0
            AC1=2500.0
            AC2=6000.0
            AC3=9500.0
            IF(DSMM-3500)120,120,122
        120 CONTINUE
            C10=3.0E02
            C11=-4.3282600E02
            C12=3.2997113E02
            C13=-1.5834140E02
            C14=5.0866847E01
            C15=-1.1216281E01
            C16=1.7136733E00
            C17=-1.8075704E-01
            C18=1.2897555E-02
            C19=-5.9370854E-04
            C110=1.5899722E-05
            C111=-1.8806100E-07
            GOTO 126
        122 CONTINUE
            C10=1.3446749E04
            C11=-4.9537759E03
            C12=7.7413409E02
            C13=-6.6525920E01
            C14=3.3974423E00
            C15=-1.0316724E-01
            C16=1.7256088E-03
            C17=-1.2269488E-05
            C18=0.0
            C19=0.0
            C110=0.0
            C111=0.0
        126 CONTINUE
```

```
      C1A=C10+C11*CDS+C12*CDS**2+C13*CDS**3+C14*CDS**4+C15*CDS**5
      C1B=C16*CDS**6+C17*CDS**7+C18*CDS**8+C19*CDS**9
      C1C=C110*CDS**10+C111*CDS**11
      C1=C1A+C1B+C1C
      IF(DSMM-800.0)128,128,129
  128 CONTINUE
      CZ0=5.8102535E-07
      CZ1=-9.3288026E-04
      CZ2=2.5001039E-03
      CZ3=-2.3495371E-03
      CZ4=1.1630954E-03
      CZ5=-3.4300115E-04
      CZ6=6.3936609E-05
      CZ7=-7.7173709E-06
      CZ8=6.0093755E-07
      CZ9=-2.9097831E-08
      CZ10=7.9588319E-10
      CZ11=-9.3862036E-12
      GOTO 132
  129 CONTINUE
      IF(DSMM-3500)140,140,142
  140 CONTINUE
      CZ0=-5.2014956E00
      CZ1=1.8290391E00
      CZ2=-2.8868848E-01
      CZ3=2.6890909E-02
      CZ4=-1.6353769E-03
      CZ5=6.7947483E-05
      CZ6=-1.9633744E-06
      CZ7=3.9404340E-08
      CZ8=-5.3807386E-10
      CZ9=4.7621674E-12
      CZ10=-2.4601875E-14
      CZ11=5.6258519E-17
      GOTO 132
  142 CONTINUE
      CZ0=6.1111277E02
      CZ1=-4.9661811E01
      CZ2=1.7522450E00
      CZ3=-3.5075306E-02
      CZ4=4.3588585E-04
      CZ5=-3.4449323E-06
      CZ6=1.6915632E-08
      CZ7=-4.7199626E-11
      CZ8=5.7320258E-14
      CZ9=0.0
      CZ10=0.0
      CZ11=0.0
  132 CONTINUE
      CC1A=CZ0+CZ1*CAS+CZ2*CAS**2+CZ3*CAS**3+CZ4*CAS**4+CZ5*CAS**5
      CC1B=CZ6*CAS**6+CZ7*CAS**7+CZ8*CAS**8+CZ9*CAS**9+CZ10*CAS**10
      CC1C=CZ11*CAS**11
      CC1=1000.0*(CC1A+CC1B+CC1C)
```

```
      BC2=0.02
      DEL=0.001
      BC1=15.0
      IF(DSMM-3500)144,144,146
  144 CONTINUE
      CD0=-7.9887704E-05
      CD1=6.3085251E-04
      CD2=-7.4214518E-06
      CD3=-1.1141749E-05
      CD4=2.3517785E-06
      CD5=-2.0060899E-07
      CD6=9.4187498E-09
      CD7=-2.6952086E-10
      CD8=4.8360439E-12
      CD9=-5.3232761E-14
      CD10=3.2890826E-16
      CD11=-8.7376943E-19
      GOTO 148
  146 CONTINUE
      CD0=-1.7802618E01
      CD1=1.2909806E00
      CD2=-3.9798780E-02
      CD3=6.7678723E-04
      CD4=-6.8501186E-06
      CD5=4.1241283E-08
      CD6=-1.3668450E-10
      CD7=1.9232534E-13
      CD8=0.0
      CD9=0.0
      CD10=0.0
      CD11=0.0
  148 CONTINUE
      C01=CD0+CD1*CAS+CD2*CAS**2+CD3*CAS**3+CD4*CAS**4+CD5*CAS**5
      C02=CD6*CAS**6+CD7*CAS**7+CD8*CAS**8+CD9*CAS**9+CD10*CAS**10
      C03=CD11*CAS**11
      C0=C01+C02+C03
      IF(DSMM-3500)150,150,152
  150 CONTINUE
      C20=1.7886330E-01
      C21=-6.4582961E-03
      C22=-1.4994404E-03
      C23=1.9047721E-04
      C24=-5.7220492E-06
      C25=-2.8715333E-07
      C26=2.8929761E-08
      C27=-1.0573124E-09
      C28=2.1306976E-11
      C29=-2.4954492E-13
      C210=1.5955587E-15
      C211=-4.3182053E-18
      GOTO 156
  152 CONTINUE
      C20=1.8711536E03
```

```
      C21=-1.3477721E02
      C22=4.1114309E00
      C23=-6.8902750E-02
      C24=6.8561083E-04
      C25=-4.0530542E-06
      C26=1.3187613E-08
      C27=-1.8228129E-11
      C28=0.000000000000000
      C29=0.000000000000000
      C210=0.00000000000000
      C211=0.00000000000000
  156 CONTINUE
      C2A=C20+C21*CAS+C22*CAS**2+C23*CAS**3+C24*CAS**4+C25*CAS**5
      C2B=C26*CAS**6+C27*CAS**7+C28*CAS**8+C29*CAS**9
      C2C=C210*CAS**10+C211*CAS**11
      C2=1000.0*(C2A+C2B+C2C)
      IF(DSMM-3500)134,134,136
  134 CONTINUE
      CK0=-7.8699233E-05
      CK1=3.2234471E-01
      CK2=1.4716139E00
      CK3=-1.1769343E00
      CK4=4.9592895E-01
      CK5=-1.3306910E-01
      CK6=2.3581299E-02
      CK7=-2.7710718E-03
      CK8=2.1279022E-04
      CK9=-1.0238109E-05
      CK10=2.7966863E-07
      CK11=-3.3061349E-09
      GOTO 138
  136 CONTINUE
      CK0=-3.1842085E05
      CK1=1.2488155E05
      CK2=-2.0808714E04
      CK3=1.9071617E03
      CK4=-1.0375390E02
      CK5=3.3493295E00
      CK6=-5.9403607E-02
      CK7=4.4664367E-04
      CK8=0.0
      CK9=0.0
      CK10=0.0
      CK11=0.0
  138 CONTINUE
      CKAP1=CK0+CK1*CDS+CK2*CDS**2+CK3*CDS**3+CK4*CDS**4+CK5*CDS**5
      CKAP2=CK6*CDS**6+CK7*CDS**7+CK8*CDS**8+CK9*CDS**9+CK10*CDS**10
      CKAP3=CK11*CDS**11
      CKAP=CKAP1+CKAP2+CKAP3
      CC2=(1.0+CKAP+CKAP**2)/3.0
      E0B=1.3051737
      C0B=0.00698505
      E0A=1.1234066
```

```
      COA=0.0103823
      FCT1=1.0
      FCT2=1.0
      FCT3=1.0
      FCT4=1.0
      FREQ=15.0
      FREQT=15.0
      FREQR=8.0
      IF(DSMM-3500)160,162,162
  160 CONTINUE
      BBF0=-5.2502975E-02
      BBF1=2.4290747E-01
      BBF2=-5.3572645E-02
      BBF3=6.6367522E-03
      BBF4=-2.850117E-04
      BBF5=-4.2239845E-06
      BBF6=8.6076538E-07
      BBF7=-3.5906169E-08
      BBF8=7.7332822E-10
      BBF9=-9.4366145E-12
      BBF10=6.2002737E-14
      BBF11=-1.7094962E-16
      GOTO 164
  162 CONTINUE
      BBF0=-1.6156112E02
      BBF1=1.2157621E01
      BBF2=-3.7874985E-01
      BBF3=6.3966447E-03
      BBF4=-6.3479456E-05
      BBF5=3.7107308E-07
      BBF6=-1.1851895E-09
      BBF7=1.5976850E-12
      BBF8=0.0
      BBF9=0.0
      BBF10=0.0
      BBF11=0.0
  164 CONTINUE
      BBFA=BBF0+BBF1*CAS+BBF2*CAS**2+BBF3*CAS**3+BBF4*CAS**4
      BBFB=BBF5*CAS**5+BBF6*CAS**6+BBF7*CAS**7+BBF8*CAS**8
      BBFC=BBF9*CAS**9+BBF10*CAS**10+BBF11*CAS**11
      BBF=BBFA+BBFB+BBFC
      IF(DSMM-3500)180,180,182
  180 CONTINUE
      EEF0=-1.6392956E-04
      EEF1=3.8696350E-02
      EEF2=-7.5764972E-03
      EEF3=9.1875738E-04
      EEF4=-7.1071983E-05
      EEF5=3.6245638E-06
      EEF6=-1.2456023E-07
      EEF7=2.8953761E-09
      EEF8=-4.4780369E-11
      EEF9=4.4053809E-13
```

```
      EEF10=-2.4905921E-15
      EEF11=6.1533305E-18
      GOTO 189
  182 CONTINUE
      IF(DSMM-4000)184,184,186
  184 CONTINUE
      EEF0=2.9869271E04
      EEF1=-1.9805576E03
      EEF2=5.2478918E01
      EEF3=-6.9458127E-01
      EEF4=4.5919943E-03
      EEF5=-1.2131410E-05
      EEF6=0.0
      EEF7=0.0
      EEF8=0.0
      EEF9=0.0
      EEF10=0.0
      EEF11=0.0
      GOTO 189
  186 CONTINUE
      EEF0=1.588114E02
      EEF1=-1.0509675E01
      EEF2=2.9687507E-01
      EEF3=-4.6404242E-03
      EEF4=4.3351686E-05
      EEF5=-2.4208222E-07
      EEF6=7.4825315E-10
      EEF7=-9.8764637E-13
      EEF8=0.0
      EEF9=0.0
      EEF10=0.0
      EEF11=0.0
  189 CONTINUE
      EEFA=EEF0+EEF1*CAS+EEF2*CAS**2+EEF3*CAS**3+EEF4*CAS**4
      EEFB=EEF5*CAS**5+EEF6*CAS**6+EEF7*CAS**7+EEF8*CAS**8
      EEFC=EEF9*CAS**9+EEF10*CAS**10+EEF11*CAS**11
      EEF=EEFA+EEFB+EEFC
      FS=16.0
      PI=3.14159265
      POIS1=0.3
      POIS2 = 0.3
      POIS3 = 0.35
      POIS4 = 0.35
      OMEGT=2.0*PI*FREQT
      OMEGR=2.0*PI*FREQR
      EMD1 = 1.3*10**6
      EMD2 = 1.3*10**6
      EMD3 = 4.0*10**4
      EMD4 = 10000
  633 EMD4 = EMD4+1000
      GMD1=EMD1/(2.0*(1.0+POIS1))
      GMD2=EMD2/(2.0*(1.0+POIS2))
      GMD3=EMD3/(2.0*(1.0+POIS3))
```

AD-A087 186 ARMY ENGINEER WATERWAYS EXPERIMENT STATION VICKSBURG--ETC F/G 1/5
PAVEMENT EVALUATION AND OVERLAY DESIGN USING VIBRATORY NONDESTR--ETC(U)
MAR 80 R A WEISS DOT-FA73WAI-377
UNCLASSIFIED FAA-RD-77-186-VOL-1 NL

2 OF 2
AD
A087186

END
DATE
FILMED
9-80
DTIC

```
      GMD4=EMD4/(2.0*(1.0+POIS4))
      EMOD1=EMD1/1.0E09
      EMOD2=EMD2/1.0E09
      EMOD3=EMD3/1.0E09
      EMOD4=EMD4/1.0E09
      SHEAR1=GMD1/1.0E09
      SHEAR2=GMD2/1.0E09
      SHEAR3=GMD3/1.0E09
      SHEAR4=GMD4/1.0E09
      Q1=(1.0-POIS1)/(1.0-2.0*POIS1)
      Q2=(1.0-POIS2)/(1.0-2.0*POIS2)
      Q3=(1.0-POIS3)/(1.0-2.0*POIS3)
      Q4=(1.0-POIS4)/(1.0-2.0*POIS4)
      S1=(1.0-POIS1)*Q1
      S2=(1.0-POIS2)*Q2
      S3=(1.0-POIS3)*Q3
      S4=(1.0-POIS4)*Q4
      H1=CC2*PI*S1*AC1/2.0
      H2=CC2*PI*S2*(AC2-AC1)/2.0
      H3=CC2*PI*S3*(AC3-AC2)/2.0
      H1 = 3.0
      H2 = 3.0
      H3 = 7.0
      HT1=H1
      HT2=H2
      HT3=H3
      H1=H1*1000
      H2=H2*1000
      H3=H3*1000
      HH2=H1+H2
      HH3=H1+H2+H3
      ARG=H1*EMOD1+H2*EMOD2+H3*EMOD3+EMOD4
  715 FORMAT(//1H ,6X,2HA1,12X,2HA2,12X,2HA3,12X,2HA4/1P4E14.6)
      OMEG=2.0*PI*FREQ
      SO=1.0/U(2)
      ALPH1=U(4)*SO**5
      ALPH1=ALPH1/1.414
      ALPH2=U(6)*SO**9
      ALPH2=ALPH2/1.6378
      W=16.0*CC1
      VM=W/AG
  655 FORMAT(/,' WE,WR,FREQT',/,1P3E14.6)
      SKO=VM*OMEG**2+SQRT((SO**2)-(CO*OMEG)**2)
      IF(FREQ-16.0)227,5,5
    5 SKO=VM*OMEG**2-SQRT((SO**2)-(CO*OMEG)**2)
  227 SK1=VM*OMEG**2
      SK2=SKO-SK1
      OMEGC=SQRT(SKO/VM)
      FREQC=OMEGC/(2.0*PI)
      FPSI=(FDYN**2)/SO**4
      ADYN=(FDYN/SO)*(1.0+ALPH1*FPSI+ALPH2*FPSI**2)
      BB=-ALPH1/(C1*(SKO-VM*OMEG**2))
      BB=-ABS(BB)
```

```
      ZZ=7./2.*(C1**2)*(BB**2)*((SKO-VM*OMEG**2)**2)-ALPH2
      WW=C2*(SKO-VM*OMEG**2)
      VV=(C1**2)*(BB**2)/(2.0*C2*(SKO-VM*OMEG**2))
      EE=(ZZ/WW)-VV
      EE=ABS(EE)
      SK=SKO+C1*BB*BBF*(ADYN**2)+C2*EE*EEF*(ADYN**4)
      STIF=SQRT((SK-VM*OMEG**2)**2+(CO*OMEG)**2)
      PS=FS/(PI*AO**2)
      POIS=POIS1
      QO=Q1
      SHEAR=SHEAR1
      DLO=PI*AO*((1.0-POIS)**2)/(2.0*(1.0-2.0*POIS))
      DLO=DLO*CC2
      IF(DLO-H1)6,6,210
    6 FKOO=2.0*PI*(AO**2)*Q1*SHEAR1/DLO
      FKOO=FKOO*CC2
      DL2=-BB*(DLO**2)/(4.0*PI*(AO**2)*QO*SHEAR)
      DL2=DL2/CC2
      CL4=DLO*(DL2/DLO)**2
      EL4=EE*(DLO**2)/(6.0*PI*(AO**2)*QO*SHEAR)
      EL4=EL4/CC2
      DL4=CL4-EL4
      BBA=-4.0*PI*(AO**2)*Q1*SHEAR1*DL2/(DLO**2)
      BBA=BBA*CC2
      DELTA=(DL2/DLO)**2-DL4/DLO
      EEA=6.0*PI*(AO**2)*Q1*SHEAR1*DELTA/DLO
      EEA=EEA*CC2
      GOTO220
  210 DLO=CC2*(S2*AO+AC1*(S1-S2))*PI/2.0
      IF(DLO-HH2)7,7,230
    7 FLO=2.0*PI*(AO**2)/DLO**2
      FKOO=FLO*(H1*(Q1*SHEAR1-Q2*SHEAR2)+Q2*SHEAR2*DLO)
      FKOO=FKOO*CC2
      FL2=2.0*H1*(Q1*SHEAR1-Q2*SHEAR2)+Q2*SHEAR2*DLO
      DL2=-BB*(DLO**3)/(4.0*PI*(AO**2)*FL2)
      DL2=DL2/CC2
      FL4=3.0*H1*(Q1*SHEAR1-Q2*SHEAR2)+Q2*SHEAR2*DLO
      DL4=((DL2**2)*FL4/DLO-EE*(DLO**3)/(6.0*CC2*PI*AO**2))/FL2
      QG12=Q1*SHEAR1-Q2*SHEAR2
      BBA=-4.0*PI*(AO**2)*DL2*(2.0*H1*QG12+Q2*SHEAR2*DLO)/(DLO**3)
      BBA=BBA*CC2
      RHO=3.0*((DL2/DLO)**2)-2.0*DL4/DLO
      DELTA=(DL2/DLO)**2-DL4/DLO
      EEA=6.0*PI*(AO**2)*(RHO*H1*QG12+DELTA*Q2*SHEAR2*DLO)/(DLO**2)
      EEA=EEA*CC2
      GOTO220
  230 DLO=CC2*(S3*AO+AC2*(S2-S3)+AC1*(S1-S2))*PI/2.0
      IF(DLO-HH3)8,8,240
    8 QG13=Q1*SHEAR1-Q3*SHEAR3
      QG23=Q2*SHEAR2-Q3*SHEAR3
      FLO=2.0*PI*(AO**2)/DLO**2
      FKOO=CC2*FLO*(H1*QG13+H2*QG23+Q3*SHEAR3*DLO)
      GL2=2.0*(H1*QG13+H2*QG23)+Q3*SHEAR3*DLO
```

```
      GL22=GL2/2.0
      DL2=-BB*(DLO**3)/(4.0*PI*(AO**2)*CC2*GL2)
      FL4=3.0*(H1*QG13+H2*QG23)+Q3*SHEAR3*DLO
      DL4=((DL2**2)*FL4/DLO-EE*(DLO**3)/(6.0*PI*CC2*AO**2))/GL2
      BBA=-4.0*PI*(AO**2)*DL2*CC2*GL2/DLO**3
      RHO=3.0*((DL2/DLO)**2)-2.0*DL4/DLO
      DELTA=(DL2/DLO)**2-DL4/DLO
      EEA1=6.0*PI*(AO**2)*CC2/DLO**2
      EEA=EEA1*(RHO*(H1*QG13+H2*QG23)+DELTA*Q3*SHEAR3*DLO)
      GOTO220
  240 DLO=CC2*(S4*AO+AC3*(S3-S4)+AC2*(S2-S3)+AC1*(S1-S2))*PI/2.0
      DLO0=-1.6182520E-01
      DLO1=2.9406204E01
      DLO2=-1.5550641E01
      DLO3=1.1928522E01
      DLO4=-4.4597082E00
      DLO5=8.8599714E-01
      DLO6=-1.0401599E-01
      DLO7=7.6323389E-03
      DLO8=-3.5492892E-04
      DLO9=1.0179406E-05
      DLO10=-1.6440288E-07
      DLO11=1.1443959E-09
      DLOA=DLO0+DLO1*CDS+DLO2*CDS**2+DLO3*CDS**3+DLO4*CDS**4
      DLOB=DLO5*CDS**5+DLO6*CDS**6+DLO7*CDS**7+DLO8*CDS**8
      DLOC=DLO9*CDS**9+DLO10*CDS**10+DLO11*CDS**11
      DLO=1000.0*(DLOA+DLOB+DLOC)
      ALPP=(CKAP-1.0)*AO/DLO
      FAD0=1.0+ALPP*DLO/AO+((ALPP*DLO/AO)**2)/3.0
      FAN1=1.0+ALPP*H1/AO+((ALPP*H1/AO)**2)/3.0
      FAB2=H1*HH2+H1**2+HH2**2
      FAN2=1.0+ALPP*(H1+HH2)/AO+((ALPP/AO)**2)*FAB2/3.0
      FAB3=HH2*HH3+HH2**2+HH3**2
      FAN3=1.0+ALPP*(HH2+HH3)/AO+((ALPP/AO)**2)*FAB3/3.0
      FAB4=HH3*DLO+HH3**2+DLO**2
      FAN4=1.0+ALPP*(HH3+DLO)/AO+((ALPP/AO)**2)*FAB4/3.0
      FAC1=FAN1/FAD0
      FAC2=FAN2/FAD0
      FAC3=FAN3/FAD0
      FAC4=FAN4/FAD0
      QG14=Q1*SHEAR1-Q4*SHEAR4
      QG24=Q2*SHEAR2-Q4*SHEAR4
      QG34=Q3*SHEAR3-Q4*SHEAR4
      FLO=2.0*PI*(AO**2)/DLO**2
      FKOO=CC2*FLO*(H1*QG14*FAC1+H2*QG24*FAC2+H3*QG34*FAC3)
      FKOO=FKOO+Q4*SHEAR4*DLO*FAC4
      GL2=2.0*(H1*QG14*FAC1+H2*QG24*FAC2+H3*QG34*FAC3)
      GL2=GL2+Q4*SHEAR4*DLO*FAC4
      DL2=-BB*(DLO**3)/(4.0*PI*(AO**2)*CC2*GL2)
      FL4=3.0*(H1*QG14*FAC1+H2*QG24*FAC2+H3*QG34*FAC3)
      FL4=FL4+Q4*SHEAR4*DLO*FAC4
      DL4=((DL2**2)*FL4/DLO-EE*(DLO**3)/(6.0*PI*CC2*AO**2))/GL2
      BBA=-4.0*PI*(AO**2)*DL2*CC2*GL2/DLO**3
```

```
      RHO=3.0*((DL2/DLO)**2)-2.0*DL4/DLO
      DELTA=(DL2/DLO)**2-DL4/DLO
      EEA1=6.0*PI*(AO**2)*CC2/DLO**2
      EEA2=RHO*(H1*QG14*FAC1+H2*QG24*FAC2+H3*QG34*FAC3)
      EEA2=EEA2+DELTA*Q4*SHEAR4*DLO*FAC4
      EEA=EEA1*EEA2
  220 CONTINUE
      ARG2=H1*EMOD1+H2*EMOD2+H3*EMOD3+(DLO-HH3)*EMOD4
C
C   THE FOLLOWING GIVES THE CALCULATION OF THE STATIC ELASTIC
C   DISPLACEMENT OF THE PAVEMENT SURFACE
C
      ZP(1)=-1.0*FS
      ZP(2)=FKOO
      ZP(3)=0.0
      ZP(4)=BBA
      ZP(5)=0.0
      ZP(6)=EEA
      CALLDOWNH(ZP,5,RR,CR)
  680 FORMAT(//5(4HROOT,I2,5X,1P2E20.7/)//
     & 5X,3H-FS,5X,1PE20.7/5X,4HFKOO,4X,1PE20.7/13X,1PE20.7/
     & 5X,3HBBA,5X,1PE20.7/13X,1PE20.7/5X,3HEEA,5X,1PE20.7)
      XE=0.0
      DO221K=1,5
      IF(CR(K))221,15,221
   15 IF(RR(K)-XE)221,221,16
   16 XE=RR(K)
C
  221 CONTINUE
      VME=VM+BC2*SKO*((BC1-FREQT)/OMEGT)**2
      WE=CC1
      VMR=VM+BC2*SKO*((BC1-FREQR)/OMEGR)**2
      WR=VMR*AG/16.0
      OMEG=OMEGT
      RBA=4.0
      REA=(1.0-CUR)*180.0+CUR*17.0
      THETA=(4.0/3.0)*C1*BBF
      ETA=(8.0/5.0)*C2*EEF
      SKOT=FKOO+3.0*RBA*BBA*(XE**2)+5.0*REA*EEA*(XE**4)
      CCC=CO
      SOT=SQRT((SKOT-VME*OMEG**2)**2+(CCC*OMEG)**2)
      SKT=SKOT+C1*BBA*BBF*(ADYN**2)+C2*EEA*EEF*(ADYN**4)
      STIFT=SQRT((SKT-VME*OMEG**2)**2+(CCC*OMEG)**2)
      FDYNT=FDYN
      FPSIT=(FDYNT**2)/SOT**4
      ADYNT=(FDYNT/SOT)*(1.0+ALPH1*FPSIT+ALPH2*FPSIT**2)
      OMETR=SQRT(SKOT/VMR)
      OM=(C1*BBF*BBA*(ADYN**2)+C2*EEF*EEA*(ADYN**4))/VMR
      OMEGTR=SQRT(OMETR**2+OM-2.0*(CO/(2.0*VMR))**2)
      FREQTR=OMEGTR/(2.0*PI)
      VMET=VM+BC2*SKOT*((BC1-FREQT)/OMEGT)**2
      WET=VMET*AG/16.0
      VMRT=VM+BC2*SKOT*((BC1-FREQTR)/OMEGTR)**2
```

```
      WRT=VMRT*AG/16.0
      SOTT=SQRT((SKOT-VMET*OMEGT**2)**2+(CCC*OMEGT)**2)
      ALPHT1=-BBA*C1*(SKOT-VMET*OMEGT**2)
      WW=C2*(SKOT-VMET*OMEGT**2)
      VV=(C1**2)*(BBA**2)/(2.0*C2*(SKOT-VMET*OMEGT**2))
      ZZ=(EEA+VV)*WW
      ALPHT2=7./2.*(C1**2)*(BBA**2)*((SKOT-VMET*OMEGT**2)**2)-ZZ
      FPSITT=(FDYNT**2)/SOTT**4
      ADYNTT=(FDYNT/SOTT)*(1.0+ALPHT1*FPSITT+ALPHT2*FPSITT**2)
      SKTT=SKOT+C1*BBF*BBA*(ADYNT**2)+C2*EEF*EEA*(ADYNTT**4)
      STIFTT=SQRT((SKTT-VMET*OMEGT**2)**2+(CCC*OMEGT)**2)
      DEAN=FDYNT/ADYNTT
      DFDYN=FDYNT+DEL
      DFPIST=(DFDYN**2)/SOTT**4
      DADYN=(DFDYN/SOTT)*(1.0+ALPHT1*DFPIST+ALPHT2*DFPIST**2)
      DSM=DEL/(DADYN-ADYNTT)
      DSMT=1000.0*DSM
      DELDSM=DSM-DSMM
      IF(DSMM-DSMT)633,733,733
  733 CONTINUE
      STIF=STIF*1000.
      STIFT=STIFT*1000.
      WRITE(6,40)
   40 FORMAT('        STIF               STIFT               DSMM        DSM',
     &'    T')
      PRINT551,STIF,STIFT,DSMM,DSMT
      WRITE(6,41)
   41 FORMAT('       EMD1         EMD2             EMD3           EMD4')
      PRINT551,EMD1,EMD2,EMD3,EMD4
      WRITE(6,42)
   42 FORMAT('       GMD1           GMD2             GMD3           GMD4')
      PRINT551,GMD1,GMD2,GMD3,GMD4
      WRITE(6,43)
   43 FORMAT('          POIS1         POIS2           POIS3        POIS4')
      PRINT551,POIS1,POIS2,POIS3,POIS4
      WRITE(6,44)
   44 FORMAT('          HT1               HT2             HT3')
      PRINT551,HT1,HT2,HT3
C
C  THIS PART GIVES A SECOND METHOD OF CALCULATING THE STATIC
C  ELASTIC DISPLACEMENT OF THE PAVEMENT SURFACE
C
      ZP(2)=SKO
      ZP(4)=-2.0*BB
      ZP(5)=-10.0*FS/SO*EE
      ZP(6)=-4.0*EE
      CALLDOWNH(ZP,5,RR,CR)
      XEE=0.0
      DO223K=1,5
      IF(CR(K))223,19,223
   19 IF(XEE-RR(K))21,223,223
   21 IF(RR(K))223,223,22
   22 XEE=RR(K)
```

```
      C                                                                 00003660
  223 CONTINUE                                                          00003665
      SKOO=FS/XEE-BB*(XEE**2)-EE*(XEE**4)                               00003670
      STOP                                                              00003680
  570 WRITE(6,45)
   45 FORMAT('PROGRAM SIZE LIMIT IS 100 DATA POINTS.')
      GO TO 628                                                         00003690
  576 WRITE(6,46)
   46 FORMAT('ELEVENTH DEGREE IS THE LIMIT.')
  592 FORMAT(F6.0)                                                      00003750
  590 READ(FNAME,592)AO                                                 00003755
      N=5                                                               00003765
      X(1)=0.0
      Y(1)=0.0
      DO600I=2,98,2                                                     00003775
  593 FORMAT(F4.0,1X,F6.2)                                              00003780
      READ(FNAME,593,END=602)X(I),Y(I)                                  00003785
      X(I+1)=-X(I)                                                      00003795
      Y(I+1)=-Y(I)                                                      00003800
  600 CONTINUE                                                          00003605
  602 M=I-1                                                             00003810
      GO TO 36                                                          00003815
  616 ITEMP=N-1
      PRINT 617,ITEMP
  617 FORMAT(/' TOO FEW POINTS FOR FITTING DEGREE',I4)
  628 WRITE(6,47)
   47 FORMAT('STOP')
      STOP
      END
      C                        SUBPROGRAM ZORP                          00003840
      C                                                                 00003845
      C                                                                 00003850
      C  THE SUBPROGRAM ZORP2 CALCULATES THE ROOTS OF THE FIFTH ORDER    00003855
      C  POLYNOMIAL THAT CONNECTS THE STATIC LOAD AND THE STATIC         00003860
      C  ELASTIC DISPLACEMENT OF THE PAVEMENT SURFACE DIRECTLY BENEATH   00003865
      C  THE VIBRATOR BASEPLATE                                         00003870
      C  ZORP2                                                          00003875
      C  ROUTINES FOR SOLVING POLYNOMIALS                               00003880
        SUBROUTINE POLY(N,A,R,C,PR,PC,RHO,PHI)                          00003885
        DIMENSION A(9999)                                               00003890
        IF(RHO)10,5,10                                                  00003895
  5     R=A(1)                                                          00003900
        C=0.                                                            00003905
        PR=A(2)                                                         00003910
        PC=0.                                                           00003915
        RETURN                                                          00003920
  10    V1=1.                                                           00003925
        V2=0.                                                           00003930
        R=A(1)                                                          00003935
        C=0.                                                            00003940
        PR=0.                                                           00003945
        PC=0.                                                           00003950
        W1=RHO*COS(PHI)                                                 00003955
```

```
      W2=RHO*SIN(PHI)
      NN=N+1
      DO 20 I=2,NN
      T1=W1*V1-W2*V2
      V2=W2*V1+W1*V2
      V1=T1
      R=R+A(I)*V1
      C=C+A(I)*V2
      PR=PR+A(I)*(I-1)*V1
   20 PC=PC+A(I)*(I-1)*V2
      PR=PR/RHO
      PC=PC/RHO
 5001 RETURN
      END
      SUBROUTINE ARCTA(X,Y,ANGLE)
      PI=3.14159265
      IF(X)10,30,20
   10 ANGLE=ATAN(Y/X)+PI*SIGN(1.,Y)
      RETURN
   20 ANGLE=ATAN(Y/X)
      RETURN
   30 IF(Y)40,60,50
   40 ANGLE=-PI/2.
      RETURN
   50 ANGLE=PI/2.
      RETURN
   60 ANGLE=0.
      RETURN
      END
      SUBROUTINE DOWNH(A,NAR,RR,CR)
      DIMENSION A(9999),RR(9999),CR(9999),Q(101),B(3)
      J=0
      N=NAR
      NPL1=N+1
      ANPP=A(NPL1)
      DO 102 I=1,NPL1
      IF (A(I))103,102,103
  102 CONTINUE
  103 C=ABS(A(I)/A(NPL1))
      LU=120
      LL=-120
      IF(C-2.**LU)100,100,101
  100 IF(C-2.**LL)101,105,105
  101 NAR=-NAR
      GO TO 5001
  105 II=(LU+LL)/2
      IF(C-2.**II)110,110,109
  109 LL=II
      GO TO 111
  110 LU=II
  111 IF(LU-LL-1)5001,112,105
  112 IB=II/N
      IF(IB)114,120,114
```

```
  114   DO 115 I=1,NPL1
        II=I-1
  115   A(I)=A(I)*(2.**(II*IB))
  120   DO 121 J1=1,NPL1
  121   A(J1)=A(J1)/A(NPL1)
  201   IF(N)2001,2001,206
  206   IF(A(1))301,211,301
 211    J=J+1
        RR(J)=0.
        CR(J)=0.
        DO 221 J1=1,N
 221    A(J1)=A(J1+1)
        N=N-1
        GO TO 201
  301   IF(N-2)601,501,401
 401    CALL GRAD(A,N,X,Y)
  421   IF(ABS(Y)-ABS(X*1.E-4))431,431,441
  431   Y=0.
  441   J=J+1
        RR(J)=X
        CR(J)=Y
        IF(Y)461,1021,461
 461    J=J+1
        RR(J)=X
        CR(J)=-Y
        GO TO 1011
 501    DISC=A(2)**2-4.*A(1)
        IF(DISC)521,541,541
  521   Y=SQRT(-DISC)/2.
        X=-A(2)/2.
        GO TO 421
 541    J=J+1
        RR(J)=(-A(2)+SQRT(DISC))/2.
        CR(J)=0.
        GO TO 1021
 601    J=J+1
        RR(J)=-A(1)
        CR(J)=0.
        GO TO 2001
 1011   B(1)=X**2+Y**2
        B(2)=-2.*X
        B(3)=1.
        NB=2
        GO TO 1041
 1021   B(1)=-RR(J)
        B(2)=1.
        NB=1
 1041   CALL DIV(A,B,N,NB,Q)
        DO 1061 J1=1,N
 1061   A(J1)=Q(J1)
        IF(CR(J))1081,1071,1081
 1071   N=N-1
        GO TO 201
```

```
 1081 N=N-2
      GO TO 201
 2001 IF(IB)2002,2005,2002
 2002 DO 2000 I=1,NAR
      RR(I)=RR(I)*(2.**(IB))
 2000 CR(I)=CR(I)*(2.**(IB))
 2005 NP1=NAR+1
      DO 2011 I=2,NP1
 2011 A(I)=0.
      A(1)=1.
      NA=0
      J=1
 2021 IF(CR(J))2041,2061,2041
 2041 NB=2
      B(3)=1.
      B(2)=-2.*RR(J)
      B(1)=RR(J)**2+CR(J)**2
      J=J+2
      GO TO 2081
 2061 NB=1
      B(2)=1.
      B(1)=-RR(J)
      J=J+1
 2081 CALL MTALGD(A,NA,B,NB,Q)
      NA=NB+NA
      NAPL1=NA+1
      DO 2091 I=1,NAPL1
 2091 A(I)=Q(I)
      IF(NA-NAR) 2021, 3001,3001
 3001 DO 3011 J2=1,NPL1
 3011 A(J2)=A(J2)*ANPP
 5001 RETURN
      END
      SUBROUTINE GRAD(A,N,XZ,YZ)
      DIMENSION A(9999),X(3),Y(3),RP(3),CP(3),RHO(3),PHI(3)
      DIMENSION ABSP(3),PR(3),PC(3)
      PI=3.14159265
      MTST=1
  101 XZ=0.0
      YZ=1.0
      DZ=2.
      RHOZ=1.
      PHIZ=PI/2.
  201 CALL POLY(N,A,RZ,CZ,PRZ,PCZ,RHOZ,PHIZ)
  221 SU=SQRT(PRZ**2+PCZ**2)
      ABSPZ=SQRT(RZ**2+CZ**2)
      U=2.*ABSPZ*SU
      PSI=ATAN(U)
      TOP=RZ*PCZ-CZ*PRZ
      BOT=-(RZ*PRZ+CZ*PCZ)
      CALL ARCTA(BOT,TOP,THETA)
      COSI=COS(THETA+PHIZ)
      SINE=SIN(THETA+PHIZ)
```

```
      IF(ABSPZ)300,5001,300
300   IF(SU)301,501,301
301   IF(RHOZ)321,401,321
321   IF(ABSPZ/(RHOZ*SU)-1.E-7)5001,5001,701
351   IF(ABSPZ/(RHOZ*SU)-10.**(-MTST))801,801,401
401   DZ=DZ/8.0
      IM=0
      DO 431 I=1,3
      DZ=2.*DZ
      X(I)=XZ+DZ*COSI
      Y(I)=YZ+DZ*SINE
      RHO(I)=SQRT(X(I)**2+Y(I)**2)
      CALL ARCTA(X(I), Y(I), PHI(I))
      CALL POLY(N,A,RP(I),CP(I),PR(I), PC(I),RHO(I),PHI(I))
      ABSP(I)=SQRT(RP(I)**2+CP(I)**2)
      IF(ABSPZ-ABSP(I)) 431,431,421
421   ABSPZ=ABSP(I)
      IM=I
431   CONTINUE
      IF(IM) 441,441,461
441   DZ=DZ/8.
      IF(RHOZ)443,445,443
443   IF(DZ/RHOZ-1.E-7)451,451,401
445   IF(DZ-1.E-7)451,451,401
451   IF(SU-ABSPZ) 501,501,5001
461   DZ=(2.**(IM-2))*DZ
      XZ=X(IM)
      YZ=Y(IM)
      PHIZ=PHI(IM)
      PRZ=PR(IM)
      PCZ=PC(IM)
      RHOZ=RHO(IM)
      RZ=RP(IM)
      CZ=CP(IM)
      GO TO 221
501   DZ=1.0
      DTHETA=PI/10.
521   THETA=0.0
      DO 561 I=1,20
      THETA=THETA+DTHETA
      XS=XZ+DZ*COS(PHIZ+THETA)
      YS=YZ+DZ*SIN(PHIZ+THETA)
      RHOS=SQRT(XS**2+YS**2)
      CALL ARCTA(XS,YS,PHIS)
      CALL POLY(N,A,RS,CS,PRS,PCS,RHOS,PHIS)
      ABSP(1)=SQRT(RS**2+CS**2)
      IF(ABSPZ-ABSP(1))561,561,601
561   CONTINUE
      DZ=DZ/2.
      IF(RHOS)563,565,563
563   IF(DZ/RHOS-1.E-7)5001,5001,521
565   IF(DZ-1.E-7)5001,5001,521
601   XZ=XS
```

```
      YZ=YS
      PHIZ=PHIS
      RHOZ=RHOS
      ABSPZ=ABSP(1)
      PRZ=PRS
      PCZ=PCS
      RZ=RS
      CZ=CS
      GO TO 221
  701 IF(PSI-1.E-6)711,711,351
  711 IF(SU-ABSPZ)501,501,351
  801 RHO(1)=RHOZ+BOT/SU**2
      IF(RHO(1))901,901,816
  816 PHI(1)=PHIZ+TOP/(RHOZ*SU**2)
  821 CALL POLY(N,A,RZ,CZ,PRZ,PCZ,RHO(1),PHI(1))
      ABSP(1)=SQRT(RZ**2+CZ**2)
      IF(ABSP(1)-ABSPZ)851,881,881
  841 XZ=RHOZ*COS(PHIZ)
      YZ=RHOZ*SIN(PHIZ)
      GO TO 5001
  851 RHOZ=RHO(1)
      ABSPZ=ABSP(1)
      PHIZ=PHI(1)
      TOP=RZ*PCZ-CZ*PRZ
      BOT=-(RZ*PRZ+CZ*PCZ)
      SU=SQRT(PRZ**2+PCZ**2)
      IF(SU)855,501,855
  855 U=2.*ABSPZ*SU
      PSI=ATAN(U)
      IF(ABSPZ/(RHOZ*SU)-10.**(-MTST))861,861,901
  861 IF(ABSPZ/(RHOZ*SU)-1.E-7)841,841,871
  871 IF(PSI-1.E-6)881,881,801
  881 IF(SU-ABSPZ)501,501,901
  901 DZ=ABSPZ/SU
      XZ=RHOZ*COS(PHIZ)
      YZ=RHOZ*SIN(PHIZ)
      MTST=MTST+1
      GO TO 201
 5001 RETURN
      END
      SUBROUTINE MTALGD(AARG,NA,BARG,NB,C)
      DIMENSION AARG(9999),BARG(9999),C(9999),A(101),B(101)
    1 NAPL1=NA+1
      DO 21 J1=1,NAPL1
   21 A(J1)=AARG(J1)
      NBPL1=NB+1
      DO 41 J1=1,NBPL1
   41 B(J1)=BARG(J1)
      NCPL1=NAPL1+NBPL1-1
      DO 91 J1=1,NCPL1
      TEMP=0.
      DO 81 J2=1,J1
      IF(J2-NAPL1) 61,61,81
```

```
 61     N2=J1-J2+1
        IF(N2-NBPL1)71,71,81
 71     TEMP=TEMP+A(J2)*B(N2)
 81     CONTINUE
        C(J1)=TEMP
 91     CONTINUE
        RETURN
        END
        SUBROUTINE DIV(A,B,NA,NB,Q)
        DIMENSION A(9999),B(9999),Q(9999)
        I1=NA-NB+1
        DO 61 J1=1,I1
  61    Q(J1)=0.
 101    KKMAX=NA-NB+1
        DO 391 KK=1,KKMAX
        K=KK-1
 201    TEMP=0.
        IF(K-1)301,211,211
 211    DO 291 JJ=1,K
        J=JJ-1
        I1=NB-K+J
        IF(I1)291,221,221
 221    I2=NA-NB-J
        TEMP=TEMP+B(I1+1)*Q(I2+1)
  291   CONTINUE
  301   I1=NA-NB-K
        I2=NA-K
  391   Q(I1+1)=A(I2+1)-TEMP
 5001   RETURN
        END
/ START ACNM=FT05F001
  9000
   2    1.2
   4    2.5
   6    3.85
   8    5.5
  10    7.5
  12    9.4
  14   11.5
/ STOP
/ EOJ
```

APPENDIX B: COMPUTER PROGRAM PAVEVAL

The computer program PAVEVAL calculates the allowable load-carrying capacity and the required overlay thickness for rigid and flexible pavements in terms of the value of the subgrade modulus that is determined from results of vibratory nondestructive testing and in terms of the elastic moduli of the pavement layers.

The copyright of much of the information provided herein is vested in Shell Research N.V. The information is published with the expressed permission of the Shell Oil Company and Koninklijke/Shell Laboratorium, Amsterdam (KSLA). Use of the BISAR computer program in any design procedure other than the procedure contained in this document is prohibited without written permission from KSLA. It may not be included in a library of programs maintained for commercial purposes by sellers of computer services nor used in any other way which results in payment of fees to the possessor of the program solely for the usage of the program. When reference is made to the results obtained with the program, acknowledgement should be given to KSLA.

DOCUMENTATION OF THE COMPUTER PROGRAM PAVEVAL

PROGRAM IDENTIFICATION

a. Program Title. WES Pavement Evaluation Program

b. Program Code Name. PAVEVAL

c. Writer. Richard A. Weiss and Ricky Austin

d. Organization. U. S. Army Engineer Waterways Experiment Station, Vicksburg, MS 39180

e. Date. July 1977

f. Source Language. Fortran IV

g. Abstract. Program calculates the allowable load-carrying capacity and required overlay thickness of rigid and flexible pavements for single-, dual-, and dual-tandem gear configurations.

ENGINEERING DOCUMENTATION

Narrative Description. The allowable load-carrying capacity and required overlay thickness of a pavement is calculated by the combined methods of vibratory nondestructive testing and layered elastic theory. For flexible pavements, a limiting vertical strain in the subgrade and a limiting tensile strain in the AC layer are used, while for rigid pavements, a limiting tensile stress at the bottom of the PCC layer is used to relate the allowable load-carrying capacity and the required overlay thickness to the pavement structure. The computer program PAVEVAL is used to implement the layered elastic theory method of pavement evaluation. This computer program is also used to calculate the allowable load-carrying capacity and the required overlay thickness of rigid and flexible pavements for single-wheel, dual-wheel, and dual-tandem-wheel loadings.

Method of Solution. For the calculation of the allowable load-carrying capacity, the computer program PAVEVAL iterates the wheel load until the operating stress and strain in the pavement are equal to a specified limiting value. For the calculation of the required overlay thickness, PAVEVAL iterates the overlay thickness until the operating stress and strain in the pavement are less than specified limiting values.

Program Capabilities. Three subroutines are used to calculate the limiting values of the stress and the strain in the pavement. An index called EKEY is used to select the type of limiting stress or strain condition that is to be used and the depth in the pavement at which the limiting stress or strain condition is applied.

Two subgrade subroutines of the program can be used to calculate the allowable load-carrying capacity and the required overlay thickness for flexible pavements. These subroutines express the limiting strain at the top of the subgrade in terms of the load repetitions and the value of the subgrade Young's modulus. The subroutine RNN gives the limiting strain at the top of the subgrade in terms of the total number of load repetitions and does not involve values of the subgrade Young's

modulus. A more accurate representation of the limiting vertical strain at the top of the subgrade is expressed in terms of the subgrade Young's modulus value and in terms of the yearly load repetition rate. This is done in subroutine FLEX. For AC pavements using subroutine FLEX, the values of the subgrade modulus are restricted to $500 < E_s < 30000$ psi. The computer program PAVEVAL automatically introduces the limiting tensile strain at the bottom of the AC layer.

For rigid pavements, the subroutine RPAL is used to describe the limiting value of the tensile stress at the bottom of the PCC layer. The limiting tensile stress is expressed in terms of the flexural strength (R), yearly load repetition number (YRN), and the pass-to-coverage ratio for each type of landing gear.

The computer program PAVEVAL can calculate the allowable load for a rigid or a flexible pavement and the required overlay thickness for a rigid or a flexible pavement. An index EKEY2 is used to select these possibilities. By selecting the proper choices of EKEY and EKEY2, the allowable load and the required overlay thickness for rigid and flexible pavements can be calculated for a number of specified limiting stress and strain conditions.

Printed Output. The printed output consists of the allowable load-carrying capacity or the required overlay thickness for a pavement. The vertical strain at the top of the subgrade and the tensile strain at the bottom of the AC layer of flexible pavements, as well as the tensile stress at the bottom of the PCC layer, are also printed out. In addition, all components of the stress and the strain at points on the layer interfaces in the pavement directly under each of the wheel positions are printed out.

Computer Equipment. The program PAVEVAL was developed on the IBM 360/65 computer.

INPUT GUIDE FOR COMPUTER
PROGRAM PAVEVAL

The input data instructions for this WES-600 CARDIN program are shown below. The meaning of these terms is explained in the following pages.

Line 1	(a)	TEXT
Line 2	(b)	NSYS
Line 3	(c)	EKEY, EKEY2
Line 4	(d)	AA, BB, RN, ALOAD, ALIN, CAREA, DSM, SWL
		or
	(d)	ES, EA, YRN, ALOAD, ALIN, CAREA, DSM, SWL, PCRATIO
		or
	(d)	DSM, FAC, YRN, R, ALOAD, ALIN, CAREA, SWL
		or
	(d)	AA, BB, RN, ATHICK, ATLIN, CAREA, DSM, SWL
		or
	(d)	ES, EA, YRN, ATHICK, ATLIN, CAREA, DSM, SWL, PCRATIO
		or
	(d)	DSM, FAC, YRN, R, ATHICK, ATLIN, CAREA, SWL
Line 5	(e)	NLAYS ISMO IRED
Line 6	(f)	E(1), NU(1), THICK(1), AK(1) or ALK(1)
Line 7	(f)	E(2), NU(2), THICK(2), AK(2) or ALK(2)
Line 8	(g)	E(NLAYS), NU(NLAYS)
Line 9	(h)	NLOAD
Line 10	(i)	LDSTRS(1), RADIUS(1), X(1), Y(1), HOSTR(1), PSI(1)
Line 11	(i)	LDSTRS(NLOAD), RADIUS(NLOAD), X(NLOAD), Y(NLOAD), HOSTR(NLOAD), PSI(NLOAD)
Line 12	(j)	NPOS
Line 13	(k)	LAYER(1), AX(1), AY(1), DEPTH(1), ETA(1)
Line 14	(k)	LAYER(NPOS), AX(NPOS), AY(NPOS), DEPTH(NPOS), ETA(NPOS)
Line 15		If another problem is desired, return to line 3 and repeat process.

The meaning and example of each card type are as follows:

Card type (a) TEXT = problem identification, maximum of 80 characters.

Example line: Line No. Identifying information

Card type (b) NSYS = number of problems to run

Example line: Line No. NSYS

Card type (c) EKEY = limiting strain and stress subroutine code

1 = calls subroutine RNN

2 = calls subroutine FLEX

3 = calls subroutine RPAL

EKEY2 = pavement problem code

0 = allowable load

1 = overlay over flexible pavement

2 = overlay over rigid pavement

Example line: Line No. EKEY EKEY2

Card type (d) if EKEY = 1, EKEY2 = 0

AA = -0.1616727

BB = -2.2150779

RN = total number of load repetitions

ALOAD = initial load, lb

ALIN = load increment, lb

CAREA = contact area (πr^2), in.^{2}

DSM = dynamic stiffness modulus, for reference

SWL = 0

Example line: Line No. AA BB RN ALOAD ALIN CAREA DSM SWL

Card type (d) if EKEY = 2, EKEY2 = 0

ES = subgrade modulus, psi

EA = asphalt modulus of existing layer

YRN = yearly load repetition number

ALOAD = initial load, lb

ALIN = load increment, lb

CAREA = contact area (πr^2), in.^{2}

DSM = dynamic stiffness modulus, for reference

SWL = 0

PCRATIO = pass-to-coverage ratio

Example line: Line No. ES EA YRN ALOAD ALIN CAREA DSM SWL PCRATIO

Card type (d) if EKEY = 3, EKEY2 = 0

DSM = dynamic stiffness modulus, for reference

FAC = pass-to-coverage ratio

YRN = yearly load repetition number

R = flexural strength, psi

ALOAD = initial load, lb

ALIN = load increment, lb

CAREA = contact area (πr^2), in.^{2}

SWL = 0

Example line: Line No. DSM FAC YRN R ALOAD ALIN CAREA SWL

Card type (d) if EKEY = 1, EKEY2 = 1

AA = -0.1616727

BB = -2.2150779

RN = total number of load repetitions

ATHICK = initial thickness, in.

ATLIN = thickness increment, in.

CAREA = contact area (πr^2), in.^{2}

DSM = dynamic stiffness modulus, for reference

SWL = load on one wheel, lb

Example line: Line No. AA BB RN ATHICK ATLIN CAREA DSM SWL

Card type (d) if EKEY = 2, EKEY2 = 1

ES = subgrade modulus, psi

EA = asphalt modulus of existing layer, psi

YRN = yearly load repetition number

ATHICK = initial thickness, in.

ATLIN = thickness increment, in.

CAREA = contact area (πr^2), in.^{2}

DSM = dynamic stiffness modulus, for reference

SWL = load on one wheel, lb

PCRATIO = pass-to-coverage ratio

Example line: Line No. ES EA YRN ATHICK ATLIN CAREA DSM SWL PCRATIO

Card type (d) if EKEY = 3, EKEY2 = 2

DSM = dynamic stiffness modulus, for reference

FAC = pass-to-coverage ratio

YRN = yearly load repetition number

R = flexural strength, psi

ATHICK = initial thickness, in.

ATLIN = thickness increment, in.

CAREA = contact area (πr^2), in.^{2}

SWL = load on one wheel, lb, for reference

Example line: Line No. DSM FAC YRN R ATHICK ATLIN CAREA SWL

Note for card type (e) and (f)

if, EKEY = 1,2

ISMO = 0

IRED = 0

all AK(i)'s = 0

if, EKEY = 3, EKEY2 = 0

ISMO = 1

IRED = 1

AK(1) = 1000

other AK(i)'s = 0

if, EKEY = 3, EKEY2 = 1

ISMO = 1

IRED = 1

AK(1) = 0

AK(2) = 1000

other AK(i)'s = 0

Card type (e) NLAYS = number of layers in pavement system

ISMO* = 0, request for rough computational procedure

1, request for smooth computational procedure

IRED = 0, AK(i) is input in card type f

1, ALK(i) is input in card type f

Example line: Line No. NLAYS ISMO IRED

* The smooth calculation procedure is more stable but less efficient than the rough procedure and is read for systems with frictionless slip between the layers or for cases when numerical instabilities are expected.

Card type (f) E(i) = modulus of layer i

NU(i) = Poisson's ratio of layer i

THICK(i)* = thickness of layer i

AK(i)** = interface compliance

or ALK(i) = reduced interface compliance

Example line: Line No. E(i) NU(i) THICK(i) AK(i) or ALK(i)

*When coding an EKEY2 = 1 or 2 problem, set THICK (1) = 1, and layer 1 is the overlay layer.

** AK(i) values are generally very small; thus, it may be more desirable to use ALK(i) where ALK(i) = $\frac{E}{1 + V_i} \cdot$ AK(i). For complete adhesion between layers i and i + 1, set AK(i) = ALK(i) = 0. For almost frictionless slip between layers, set $\frac{E_i}{1 + V_i} \cdot$ ALK(i) $\geq$ 1000.

Card type (g) E(NLAYS) = modulus of last layer

NU(NLAYS) = Poisson's ratio of last layer

Example line: Line No. E(NLAYS) NU(NLAYS)

Card type (h) NLOAD* = number of loaded areas

Example line: Line No. NLOAD

*Single wheel enter 1, dual wheel enter 2, dual-tandem enter 4

Card type (i) Load information: one card for each load

LDSTRS(i) = vertical load in units of load for loaded area i

RADIUS(i) = radius of loaded area i

X(i)* = abscissa of center of loaded area

Y(i)* = ordinate of center of loaded area

HOSTR(i) = horizontal load in units of load for loaded area i (normally zero)

PSI(i) = angle of HOSTR(i) with respect to positive X-axis in degrees (normally zero)

Example line: Line No. LDSTRS(i) RADIUS(i) X(i) Y(i) HOSTR(i) PSI(i)

*X(i) and Y(i) should always be zero.

Card type (j) NPOS = number of depths that will be used for iteration purposes

1, rigid pavement (EKEY = 3)

2, flexible pavement (EKEY = 1 or 2)

Example line: Line No. NPOS

Card type (k) LAYER (i)* = layer number for position i

AX(i) = abscissa of position (always zero)

AY(i) = ordinate of position (always zero)

DEPTH(i)* = depth from pavement surface to position

ETA(i) = angle from which position is observed with respect to the difference of the tangential loading (always zero)

Example line: Line No. LAYER(i) AX(i) AY(i) DEPTH(i) ETA(i)

*if, EKEY = 3, EKEY2 = 0

LAYER(1) = 1

DEPTH(1) = THICK(1)

if, EKEY = 3, EKEY2 = 2

LAYER(1) = 2

DEPTH(1) = THICK(1) + THICK(2), when THICK(1) = 1

if, EKEY = 1,2, EKEY2 = 0

LAYER(1) = 1

LAYER(2) = last layer (NLAYS)

DEPTH(1) = THICK(1)

DEPTH(2)* = distance from pavement surface to top of subgrade

if, EKEY = 1,2, EKEY2 = 1

LAYER(1) = 2

LAYER(2) = last layer (NLAYS)

DEPTH(1) = THICK(1) + THICK(2), where THICK(1) = 1

DEPTH(2)* = distance from pavement surface to top of subgrade

$$*\text{DEPTH}(2) = \sum_{i=1}^{(\text{NLAYS}-1)} \text{THICK}(i)$$

PROGRAM LISTING

Data are coded, then typed into a disc file, and saved for later execution. The following listing is the sample problem with control cards.

```
      PROGRAM PAVEVL
C                        COMPUTATION OF STRESSES,STRAINS AND       00010000
C                        DISPLACEMENTS IN LAYERED ELASTIC SYSTEMS  00010010
C                        THIS PROGRAM CALCULATES THE FOLLOWING     00010020
C                        STRESSES,STRAINS AND DISPLACEMENTS        00010030
C                         1)                  RADIAL      DISPLACEMENT 00010040
C                         2)                  TANGENTIAL DISPLACEMENT 00010050
C                         3)                  VERTICAL    DISPLACEMENTS00010060
C                         4)                  RADIAL      STRESS       00010070
C                         5)                  TANGENTIAL STRESS        00010080
C                         6)                  VERTICAL    STRESS       00010090
C                         7)     RADIAL AND TANGENTIAL STRESS          00010100
C                         8)     RADIAL AND VERTICAL    STRESS         00010110
C                         9)TANGENTIAL AND VERTICAL    STRESS          00010120
C                        10)                  RADIAL      STRAIN       00010130
C                        11)                  TANGENTIAL STRAIN        00010140
C                        12)                  VERTICAL    STRAIN       00010150
C                        13)     RADIAL AND TANGENTIAL STRAIN          00010160
C                        14)     RADIAL AND VERTICAL    STRAIN         00010170
C                        15)TANGENTIAL AND VERTICAL    STRAIN          00010180
C                                                                  00010190
C                        MASTERPROGRAM                             00010200
C                                                                  00010210
C                        PURPOSE                                   00010220
C                                                                  00010230
C                          THIS MASTERPROGRAM READS DATA WHICH     00010240
C                          DETERMINE THE PHYSICAL BEHAVIOUR OF     00010250
C                          THE SYSTEM OF LAYERS AND WHICH          00010260
C                          DESCRIBE THE CONFIGURATION OF THE LOADS.00010270
C                          FOR EACH SYSTEM THE REQUIRED STRESSES   00010280
C                          STRAINS AND DISPLACEMENTS ARE READ IN.  00010290
C                          THEN THE COORDINATES OF EACH POSITION   00010300
C                          ARE READ.                               00010310
C                          FOR A COMPLETE INPUT-DESCRIPTION SEE    00010320
C                          GROUP EXTERNAL REPORT AMSR.     .72     00010330
C                        SYSTEM DATA ARE OUTPUTTED BY              00010340
C                                  1) SYSTEM                       00010350
C                        AFTER SUBSEQUENT CALLING IN OF*           00010360
C                                 2) MACON1                        00010370
C                                 4) MA2CON                        00010380
C                                 5) CONPNT                        00010390
C                                 6) ASYMPT                        00010400
C                                 7) GENDAT                        00010410
C                                 8) INGRAL                        00010420
C                        THE STRESSES,STRAINS AND DISPLACEMENTS    00010430
C                        ARE CALCULATED AND AFTER SUBSEQUENT       00010440
C                        CALLING IN OF*                            00010450
C                                 9) CALC                          00010460
C                                10)OUTPUT                         00010470
C                                11)JACOBI                         00010480
C                                12)ESORT                          00010490
```

```
C                                MAIN OUTPUTS OR HAS ALREADY OUTPUTTED#     00010500
C                               -ALL STRESSES,STRAINS AND DISPLACEMENTS,    00010510
C                                INDUCED BY EACH LOAD SEPARATELY AND        00010520
C                                EXPRESSED IN CYLINDRICAL COMPONENTS.       00010530
C                               -ALL TOTAL STRESSES STRAINS AND DISPLACE-   00010540
C                                MENTS EXPRESSED IN CARTESIAN COMPONENTS.   00010550
C                               -ALL PRINCIPAL TOTAL STRESSES AND STRAINS,00010560
C                                WITH THEIR PRINCIPLE DIRECTIONS.           00010570
C                               -ALL MAXIMUM TOTAL SHEAR STRESSES AND       00010580
C                                STRAINS,WITH THEIR PRINCIPLE DIRECTIONS    00010590
C                               -THE MIDPOINTS OF THE THREE ACCOMPANYING    00010600
C                                MOHR'S CIRCLES.                            00010610
C                               -THE TOTAL STRAIN ENERGY AND STRAIN         00010620
C                                ENERGY OF DISTORTION.                      00010630
C-----------------------------------------------------------------------00010640
C                                                                           00010650
C*********************************************************************** 00010660
C     THIS PROGRAM WAS MODIFIED TO SOLVE RIGID AND FLEXIBLE PAVEMENT        00010670
C     PROBLEMS USING THE COMBINED METHODS OF THE BISAR PROGRAM AND          00010680
C     VIBRATORY NONDESTRUCTIVE TESTING. THIS PROGRAM SOLVES OVERLAY         00010690
C     THICKNESS AND ALLOWABLE LOAD PROBLEMS FOR EACH PAVEMENT TYPE.         00010700
C                                                                           00010710
C                                                                           00010720
C               PROGRAM NAME: PAVEVAL                                       00010730
C               CODED BY: RICKY AUSTIN, GEOTECHNICAL LABORATORY,            00010740
C                         WATERWAYS EXPERIMENT STATION,VICKSBURG,           00010750
C                         MISSISSIPPI                                       00010760
C               COMPUTER: WATERWAYS EXPERIMENT STATION, GE600               00010770
C               LANGUAGE: FORTRAN IV                                        00010780
C               DATE COMPLETED: SEPTEMBER 1978                              00010790
C               SPECIAL REQUIREMENTS: CARDIN,                               00010800
C                                     REMOTE BATCH PROCESSING               00010801
C               STORAGE: DISC                                               00010810
C                                                                           00010820
C                                                                           00010830
C***********************************************************************00010840
C                                                                           00010850
      LOGICAL STRESS,EPS,RLOW,AID(27),N,L,N2,L2,NZEP,NZEQ                   00010860
      INTEGER REQEST(27),IQ(3),DATE(3),ISTRSS(27),INTV(10),IVER1(7),        00010870
     +IVER2(10)                                                             00010880
      REAL NU,K5,MU,LDSTRS(10),HOSTR(10),LOAD,INT(17),V(15),X(10),Y(10),00010890
     +A(3,3),HH(3,3),W(3),C(39),B(3,3),TEXT(15),ACCUR(3),PSI(10),AK(9),  00010900
     +ALK(9)                                                                00010910
      DOUBLE PRECISION CZ,ELLE,ELLK                                         00010920
      COMMON/ASDT/LAYER,NLAYS,M,R,Z,NU(10),ACCUR,LOAD,HOSTRS,NZEROS,H(9)00010930
     +,K5(10),E(10),AL(9),THICK(9),RADIUS(10)                               00010940
      COMMON/STRDTA/STRESS(27),EPS(17),RLOW,ST,CT,L,ACC                     00010950
      COMMON/CONST/CZ,ELLE,ELLK,ALMBDA                                      00010960
      COMMON/CNTING/F10M1,F100,F101,F11M2,F11M1,F110,F111                   00010970
      COMMON/TAPE/NOUT                                                      00010980
      COMMON VSTR,STRL,ITER,STRL2                                           00010990
      COMMON/RADIAL/STSL,DSM,FS,SWL                                         00011000
      DIMENSION XTEMP(100),YTEMP(100),LAY(100),AXX(100)                     00011010
```

```
      DIMENSION AYY(100),DEP(100),ETAA(100)                             00011020
      INTEGER EKEY,EKEY2                                                00011030
      DATA NBLANK,ISTRSS,IREF1,IREF2/                                   00011040
     +'   ','UR ','UT ','UZ ','SRR','STT','SZZ','SRT','SRZ','STZ','ERR'00011050
     +,'ETT','EZZ','ERT','ERZ','ETZ','UX ','UY ','SXX','SXY','SXZ','SYY'00011060
     +,'SYZ','EXX','EXY','EXZ','EYY','EY','LOAD','STRS'/                00011070
      DATA REQEST/'UR ','UT ','UZ ','SRR','STT','SZZ','SRT','SRZ','STZ',00011080
     1'ERR',                                                            00011090
     2'ETT','EZZ','ERT','ERZ','ETZ','UX ','UY ','SXX','SXY','SXZ','SYY',00011100
     3'SYZ','EXX','EXY','EXZ','EYY','EY '/                              00011110
      DATA IVER1,IVER2/1,2,3,6,7,13,14,4,5,8,9,10,11,12,15,16,17/       00011120
      DATA IDENT/'LOAD'/                                                00011130
C-----------------------------------------------------------------------00011140
C                                THESE ARE THREE ACCURACIES             00011150
C                                ACCUR(1) IS USED FOR TESTING SEVERAL   00011160
C                                         VARIABLES AGAINST EACH OTHER. 00011170
C                                ACCUR(2) IS USED FOR ABSOLUTE ACCURACY 00011180
C                                         OF THE INTEGRATION PROCEDURE  00011190
C                                ACCUR(3) IS USED FOR RELATIVE ACCURACY 00011200
C                                         OF THE INTEGRATION PROCEDURE  00011210
C                                NIN ,                                  00011220
C                                NOUT     ARE SYMBOLIC NAMES FOR INPUT AND 00011230
C                                         OUTPUT MEDIA RESP.            00011240
C-----------------------------------------------------------------------00011250
      ACCUR(1)=1.0E-04                                                  00011260
      ACCUR(2)=1.0E-4                                                   00011270
      ACCUR(3)=1.0E-3                                                   00011280
      ACC=ACCUR(1)                                                      00011290
      NIN=5                                                             00011300
      NOUT=6                                                            00011310
      V2=1.414214                                                       00011320
C     ITER = 0, INPUT NOT COMPLETE.                                     00011330
C     ITER = 1, INPUT COMPLETE.                                         00011340
C     ISKIP = 0, ITERATION NOT COMPLETE.                                00011350
C     ISKIP = 1, ITERATION COMPLETE.                                    00011360
      WRITE(NOUT,9000)                                                  00011370
C-----------------------------------------------------------------------00011380
C                                READ TEXT AND DATE CARD                00011390
C-----------------------------------------------------------------------00011400
      READ(NIN,9010)TEXT                                                00011410
      WRITE(NOUT,9020)TEXT                                              00011420
C-----------------------------------------------------------------------00011430
C                                READ NUMBER OF SYSTEMS AND SET LOOP    00011440
C-----------------------------------------------------------------------00011450
      CALL NFRD(NIN,NSYS,1)
      CALL NFRD(NIN,IO,2)
      DO 460 ISYS=1,NSYS                                                00011480
      ITER=0                                                            00011490
      ISKIP=0                                                           00011500
C     SELECT LIMITING STRAIN OR STRESS SUBROUTINE CODE AND PAVEMENT     00011510
C     TYPE.                                                             00011520
      CALL NFRD(NIN,EKEY,1)
      CALL NFRD(NIN,EKEY2,2)
```

```
      GO TO(501,502,503),EKEY                                           00011540
  501 CALL RNN(AA,BB,RN,ALOAD,ALIN,CAREA)                               00011550
      GO TO 510                                                         00011560
  502 CALL FLEX(ES,EA,ALOAD,ALIN,CAREA,XS,AS,BS,YRN,PRATIO)             +0011570
      GO TO 510                                                         00011580
  503 CALL RPAL(ALOAD,ALIN,CAREA)                                       00011590
  510 CONTINUE                                                          00011600
      IF(EKEY2.GT.0)GO TO 520                                           00011610
      IF(EKEY.LT.3)WRITE(NOUT,900)STRL,STRL2,ALOAD,ALIN                 00011620
      IF(EKEY.EQ.3)WRITE(NOUT,901)FS,DSM,ALOAD,ALIN,STSL                00011630
      GO TO 550                                                         00011640
  520 CONTINUE                                                          00011650
      ATHICK=ALOAD                                                      00011660
      ATLIN=ALIN                                                        00011670
      GO TO(521,522),EKEY2                                              00011680
  521 WRITE(NOUT,902)                                                   00011690
      WRITE(NOUT,903)DSM,SWL,ATHICK,ATLIN,STRL,STRL2                    00011700
      GO TO 550                                                         00011710
  522 WRITE(NOUT,904)                                                   00011720
      WRITE(NOUT,905)DSM,SWL,FS,ATHICK,ATLIN,STSL                       00011730
  550 CONTINUE                                                          00011740
C-----------------------------------------------------------------------00011750
C                                READ NUMBER OF LAYERS AND THEIR PARAMETERS00011760
C-----------------------------------------------------------------------00011770
      CALL NFRD(NIN,NLAYS,1)
      CALL NFRD(NIN,ISMO,2)
      CALL NFRD(NIN,IRED,2)
         IF(NLAYS.EQ.1)  GO TO 10                                       00011790
         M=NLAYS-1                                                      00011800
      DO 315 I=1,M                                                      00011810
      CALL FFRD(NIN,E(I),1)
      CALL FFRD(NIN,NU(I),2)
      CALL FFRD(NIN,THICK(I),2)
  315 CALL FFRD(NIN,AK(I),2)
   10 CALL FFRD(NIN,E(NLAYS),1)
      CALL FFRD(NIN,NU(NLAYS),2)
C-----------------------------------------  ------------------------00011840
C                                READ NUMBER OF LOADS AND THEIR PARAMETERS 00011850
C-----------------------------------------------------------------------00011860
      CALL NFRD(NIN,NLOAD,1)
         NZEP = .FALSE.                                                 00011880
         NZEQ = .FALSE.                                                 00011890
         DO 30 I=1,NLOAD                                                00011900
      CALL FFRD(NIN,LDSTRS(I),1)
      CALL FFRD(NIN,RADIUS(I),2)
      CALL FFRD(NIN,X(I),2)
      CALL FFRD(NIN,Y(I),2)
      CALL FFRD(NIN,HOSTR(I),2)
      CALL FFRD(NIN,PSI(I),2)
      IF(EKEY2.GT.0)GO TO 560                                           00011920
      LDSTRS(I)=ALOAD                                                   00011930
  560 CONTINUE                                                          00011940
            PSI(I)=.0174533*PSI(I)                                      00011950
```

```
            IF(LDSTRS(I).GT.ACCUR(1))  NZEP = .TRUE.                    00011960
            IF(HOSTR(I).GT.ACCUR(1))  NZEQ = .TRUE.                     00011970
            IF(IDENT.EQ.IREF1)  GO TO 20                                00011980
            IF(IDENT.NE.IREF2)  WRITE(NOUT,9040) LDSTRS(I),HOSTR(I)     00011990
            GO TO 30                                                    00012000
   20       LDSTRS(I) = LDSTRS(I)/(3.14159*RADIUS(I)*RADIUS(I))         00012010
            HOSTR(I) = HOSTR(I)/(3.14159*RADIUS(I)*RADIUS(I))           00012020
   30    CONTINUE                                                       00012030
C-----------------------------------------------------------------------00012040
C                             TEST ON OBVIOUS MISTAKES IN SYSTEM'S DATA-00012050
C                             CARDS.                                    00012060
C                             WHEN IRED > 0 THE REDUCED SPRINGCOMPLIAN- 00012070
C                             CE WAS READ.                              00012080
C                             A NON-VANISHING SLIPRESISTANCE IS SUBSTI- 00012090
C                             TUTED TO PREVENT RIGID-BODY MOTION OF THE 00012100
C                             TOPLAYERS                                 00012110
C-----------------------------------------------------------------------00012120
         DO 50 J = 1,NLAYS                                              00012130
            IF((1.0-NU(J)).LT.ACCUR(1)) GO TO 410                       00012140
            IF(E(J).LT.ACCUR(1)) GO TO 420                              00012150
            IF(J.EQ.NLAYS) GO TO 50                                     00012160
            IF(IRED.EQ.0) GO TO 40                                      00012170
            ALK(J) = AK(J)                                              00012180
            IF(ALK(J).LT.1000.0.OR..NOT.NZEQ) GO TO 50                  00012190
            ALK(J) = 1000.0                                             00012200
            AK(J) = 1000.0                                              00012210
            GO TO 50                                                    00012220
   40       ALK(J) = AK(J)*E(J)/(1.0+NU(J))                             00012230
            IF(ALK(J).LT.1000.0.OR..NOT.NZEQ) GO TO 50                  00012240
            ALK(J) = 1000.0                                             00012250
            AK(J) = ALK(J)*(1.0+NU(J))/E(J)                             00012260
   50    CONTINUE                                                       00012270
C-----------------------------------------------------------------------00012280
C                             OUTPUT OF ALL PHYSICAL DATA OF SYSTEM     00012290
C                             AND LOADS BY CALLING IN SYSTEM.           00012300
C-----------------------------------------------------------------------00012310
         CALL SYSTEM(ISYS,E,NU,THICK,AK,NLAYS,M,NLOAD,LDSTRS,HOSTR,ALK, 00012320
     +   RADIUS,X,Y,PSI,ISMO,IRED)                                      00012330
         IF(.NOT.NZEP.AND..NOT.NZEQ)  GO TO 430                         00012340
C-----------------------------------------------------------------------00012350
C                             CALCULATION OF CONSTANTS USED IN SUBROU-  00012360
C                             TINE MATRIX TO BUILT UP VARIOUS MATRICES  00012370
C                             BY CALLING IN MACON1.                     00012380
C-----------------------------------------------------------------------00012390
         CALL MACON1(ISMO,ALK,NEWSYS)                                   00012400
   60 IF(NEWSYS.EQ.0)  GO TO 70                                         00012410
         CALL SYSTEM(ISYS,E,NU,THICK,AK,NLAYS,M,NLOAD,LDSTRS,HOSTR,ALK, 00012420
     +   RADIUS,X,Y,PSI,ISMO,IRED)                                      00012430
C-----------------------------------------------------------------------00012440
C                             READ STRESSES,STRAINS AND DISPLACEMENTS   00012450
C                             TO BE CALCULATED.                         00012460
C-----------------------------------------------------------------------00012470
  70  CONTINUE                                                          00012480
```

```
          DO 90 I=1,27
             IF(REQEST(I).EQ.NBLANK)  GO TO 80
             IF(REQEST(I).NE.ISTRSS(I))  WRITE(NOUT,9070) ISTRSS(I)
             AID(I)=.TRUE.
             GO TO 90
   80        AID(I)=.FALSE.
   90     CONTINUE
C-----------------------------------------------------------------------
C                             CONSYS DETERMINES FOR EACH SYSTEM WHICH
C                             STRESSES,STRAINS AND DISPLACEMENTS WILL
C                             BE CALCULATED.
C-----------------------------------------------------------------------
          CALL CONSYS(AID,NZEP,NZEQ,N,L)
C-----------------------------------------------------------------------
C                             READ NUMBER OF POSITIONS AND SET LOOP
C-----------------------------------------------------------------------
  100 CALL NFRD(NIN,NPOS,1)
 590  CONTINUE
 580  CONTINUE
      IF(ISKIP.EQ.0)GO TO 88
      NPOS=NPOS3
 88   CONTINUE
          DO 400 IPOS=1,NPOS
             N2 = N
             L2 = L
             DO 110 I=1,3
             DO 110 J=1,3
  110        A(I,J)=0.0
C-----------------------------------------------------------------------
C                             READ POINT COORDINATES AND LAYERNUMBER.
C-----------------------------------------------------------------------
      IF(ITER.GT.0)GO TO 570
      CALL NFRD(NIN,LAY(IPOS),1)
      CALL FFRD(NIN,AXX(IPOS),2)
      CALL FFRD(NIN,AYY(IPOS),2)
      CALL FFRD(NIN,DEP(IPOS),2)
      CALL FFRD(NIN,ETAA(IPOS),2)
 570  CONTINUE
      XTEMP(IPOS)=AXX(IPOS)
      YTEMP(IPOS)=AYY(IPOS)
      LAYER=LAY(IPOS)
      AX=AXX(IPOS)
      AY=AYY(IPOS)
      DEPTH=DEP(IPOS)
      ETA=ETAA(IPOS)
             ETA=.0174533*ETA
             IF(NLAYS.EQ.1)  LAYER=1
             WRITE(NOUT,9090) IPOS,LAYER,AX,AY,DEPTH
             TMIN=1.0E+10
             IF(NLAYS.EQ.1)  GO TO 130
             J=LAYER+1
             J=MIN0(J,M)
             DO 120 I=1,J
```

```
            IF(THICK(I) LT.TMIN)  TMIN=THICK(I)
  120     CONTINUE
  130     UX=0.0
          UY=0.0
          UZ=0.0
          MU=NU(LAYER)
          FT=(1.0+MU)/E(LAYER)
C---------------------------------------------------------------------------
C                                 SET LOOP FOR NUMBER OF LOADS.
C---------------------------------------------------------------------------
          DO 330 I=1,NLOAD
            DO 140 J=1,17
  140       INT(J)=0.0
            DO 150 J=1,27
  150       STRESS(J)=AID(J)
C      COMPUTES THE LIMITING ASPHALT STRAIN AND SUBGRADE STRAIN.
            IF(NLAYS.EQ.1)  GO TO 160
C---------------------------------------------------------------------------
C                                 CALCULATION OF CONSTANTS NEEDED FOR THE
C                                 EVALUATION OF THE CHARACTERISTIC FUNCTI-
C                                 ONS IN MATRIX BY CALLING IN MA2CON.
C                                 MA2CON.
C---------------------------------------------------------------------------
            CALL MA2CON(TMIN,I,ISMO,ALK)
C---------------------------------------------------------------------------
C                                 DETERMINATION OF POINT COORDINATES IN THE
C                                 CYLINDRICAL COORDINATE SYSTEM WITH LOAD-
C                                 AXIS AS AXIS OF SYMMETRY.
C---------------------------------------------------------------------------
  160       IF(X(I).EQ.AX.AND.Y(I).EQ.AY)  GO TO 170
            THETA=ATAN2((AY-Y(I)),(AX-X(I)))-PSI(I)
            GO TO 180
  170       THETA=ETA-PSI(I)
  180       RADDIS=SQRT((AX-X(I))**2+(AY-Y(I))**2)
            WRITE(NOUT,9100) I,RADDIS,THETA
            R=RADDIS/RADIUS(I)
            Z=DEPTH/RADIUS(I)
            IF(NLAYS.EQ.1)  GO TO 230
  190       IF(LAYER.GT.1)  GO TO 210
            IF(Z.GT.-ACCUR(1).AND.Z.LT.(H(1)+ACCUR(1)))  GO TO 230
  200       WRITE(NOUT,9110)
            GO TO 400
  210       IF(LAYER LT.NLAYS)  GO TO 220
            IF(Z.GT.(H(M)-ACCUR(1)))  GO TO 230
            GO TO 200
  220       IF(Z.GT.(H(LAYER-1)-ACCUR(1)).AND.Z.LT.(H(LAYER)+ACCUR(1)
     +      ))  GO TO 230
            GO TO 200
  230       RADI=RADIUS(I)
            LOAD=LDSTRS(I)
            HOSTRS=HOSTR(I)
            RLOW=R.LT.ACCUR(1)
            ST=SIN(THETA)
```

```
                  CT=COS(THETA)
      C-----------------------------------------------------------------
      C                             CONPNT DETERMINES FOR EACH POINT-LOAD
      C                             CONFIGURATION WHICH INTEGRALS HAVE TO BE
      C                             CALCULATED.
      C-----------------------------------------------------------------
                  CALL CONPNT(R,HOSTRS,LOAD,Z,N2,L2)
                  IF(LAYER.NE.1)  GO TO 250
                  CZ = DBLE(Z)
                  IF(Z.LT.ACCUR(1).AND.ABS(R-1.0).LT.ACCUR(1))  GO TO 240
      C-----------------------------------------------------------------
      C                             ASYMPT DETERMINES THE LIPSCHITZ-HANKEL
      C                             INTEGRALS NEEDED FOR THE ASYMPTOTIC PART
      C                             OF THE INTEGRALS,FOR POINTS IN THE TOP-
      C                             LAYER ONLY.
      C-----------------------------------------------------------------
                  CALL ASYMPT(R,ACCUR(1))
                  GO TO 250
      C-----------------------------------------------------------------
      C                             FOR POINTS AT THE RIM OF THE LOAD THE
      C                             LIPSCHITZ-MANKEL INTEGRALS CAN BE GIVEN
      C                             DIRECTLY.
      C-----------------------------------------------------------------
        240       F10M1 = 0.63662
                  F100 = 0.5
                  F11M1 = 0.5
                  F11M2 = 0.424413
                  F101 = 0.0
                  F110 = 0.0
                  F111 = 0.0
      C-----------------------------------------------------------------
      C                             COMPUTATION OF THE REQUIRED INTEGRALS BY
      C                             CALLING IN GENDAT AND INGRAL
      C-----------------------------------------------------------------
        250       INTT = 0
                  DO 260 J = 1,17
                     INT(J) = 0.0
        260       CONTINUE
                  DO 270 J = 1,10
                     INTV(J) = 0
                     K = IVER2(J)
                     IF(.NOT.EPS(K))  GO TO 270
                     INTV(J) = K
                     INTT = INTT+1
        270       CONTINUE
                  IF(INTT.EQ.0)  GO TO 280
                  IF(NLAYS.NE.1)  CALL GENDAT(1,NZEROS,R,ACC)
                  CALL INGRAL(2,INTV,INTT,INT)
        280       INTT = 0
                  DO 290 J = 1,7
                     INTV(J) = 0
                     K = IVER1(J)
                     IF(.NOT.EPS(K))  GO TO 290
```

```
                 INTV(J) = K
              INTT = INTT+1
   290        CONTINUE
              IF(INTT.EQ.0)  GO TO 300
              IF(NLAYS.NE.1)  CALL GENDAT(0,NZEROS,R,ACC)
              CALL INGRAL(1,INTV,INTT,INT)
   300        PSI0 = PSI(I)
C-------------------------------------------------------------------------
C                            CALC COMPUTES AND OUTPUTS THE STRESSES,
C                            STRAINS AND DISPLACEMENTS,INDUCED BY EACH
C                            LOAD SEPARATEL.
C-------------------------------------------------------------------------
              CALL CALC(INT,V,R,MU,RADI,FT,LOAD,HOSTRS,PSI0,Z)
              IF(.NOT.N2)  GO TO 330
C-------------------------------------------------------------------------
C                            COMPUTATION AND SUMMATION OF CARTESIAN
C                            COORDINATES.THE USED COORDINATE SYSTEM IS
C                            THE ONE WHEREIN POINTCOORDINATES WERE
C                            STATED.
C-------------------------------------------------------------------------
              UZ     =UZ+V(3)
              IF(ABS(RADDIS).LT.ACCUR(1))  GO TO 310
              CT     = (AX-X(I))/RADDIS
              ST     = (AY-Y(I))/RADDIS
              GO TO 320
   310        CT     =COS(ETA)
              ST     =SIN(ETA)
   320        CT2    =CT*CT
              ST2    =ST*ST
              STCT   =ST*CT
              A(1,1)=A(1,1)+V(4)*CT2+V(5)*ST2-2.0*V(7)*STCT
              A(1,2)=A(1,2)+V(7)*(CT2-ST2)+(V(4)-V(5))*STCT
              A(1,3)=A(1,3)+V(8)*CT-V(9)*ST
              A(2,1)=A(1,2)
              A(2,2)=A(2,2)+V(4)*ST2+V(5)*CT2+2.0*V(7)*STCT
              A(2,3)=A(2,3)+V(8)*ST+V(9)*CT
              A(3,1)=A(1,3)
              A(3,2)=A(2,3)
              A(3,3)=A(3,3)+V(6)
              UX     =UX+V(1)*CT-V(2)*ST
              UY     =UY+V(1)*ST+V(2)*CT
   330     CONTINUE
           TRACE=A(1,1)+A(2,2)+A(3,3)
           AB    =(1.0+MU)/E(LAYER)
           AC    =MU*TRACE/E(LAYER)
           DO 350 I=1,3
              DO 340 J=1,3
                 B(I,J)=AB*A(I,J)
                 IF(I.NE.J)  GO TO 340
                 B(I,J)=B(I,J)-AC
   340        CONTINUE
   350     CONTINUE
C-------------------------------------------------------------------------
```

```
C                                        OUTPUT FOR TOTAL STRESSES,STRAINS AND
C                                        DISPLACEMENTS BY THREE TIMES CALLING IN
C                                        OUTPUT.
C-------------------------------------------------------------------------------
      IF(ISKIP.EQ.0)GO TO 363
           WRITE(NOUT,9120)
  363 CONTINUE
           EPS(1)=STRESS(18)
           EPS(2)=STRESS(21)
           EPS(3)=STRESS( 6)
           EPS(4)=STRESS(22)
           EPS(5)=STRESS(20)
           EPS(6)=STRESS(19)
             C(1)=A(1,1)
             C(2)=A(2,2)
             C(3)=A(3,3)
             C(4)=A(2,3)
             C(5)=A(1,3)
             C(6)=A(1,2)
      IF(ISKIP.EQ.0)GO TO 361
           CALL OUTPUT(EPS,C,6,1)
  361 CONTINUE
           EPS(1)=STRESS(23)
           EPS(2)=STRESS(26)
           EPS(3)=STRESS(12)
           EPS(4)=STRESS(27)
           EPS(5)=STRESS(25)
           EPS(6)=STRESS(24)
             C(1)=B(1,1)
             C(2)=B(2,2)
             C(3)=B(3,3)
             C(4)=B(2,3)
             C(5)=B(1,3)
             C(6)=B(1,2)
      IF(ISKIP.EQ.0)GO TO 362
           CALL OUTPUT(EPS,C,6,2)
  362 CONTINUE
           EPS(1)=STRESS(16)
           EPS(2)=STRESS(17)
           EPS(3)=STRESS( 3)
             C(1)=UX
             C(2)=UY
             C(3)=UZ
      IF(ISKIP.EQ.0)GO TO 360
           CALL OUTPUT(EPS,C,3,3)
   360     IF(.NOT.L2)  GO TO 400
C-------------------------------------------------------------------------------
C                                        JACOBI COMPUTES PRINCIPAL VALUES AND
C                                        DIRECTIONS OF TOTAL STRESSES AND STRAINS.
C                                        THE PRINCIPAL VALUES ARE SORTED ACCORDING
C                                        TO MAGNITUDE BY CALLING IN ESORT.
C-------------------------------------------------------------------------------
           CALL JACOBI(A,HH,3,3,1,W,IQ)
```

```
                CALL ESORT(A,HH,3,3,1,W,IQ)
      C----------------------------------------------------------------------
      C                                DETERMINATION OF MAX.SHEAR STRESSES AND
      C                                STRAINS WITH THEIR DIRECTIONS AND DETERMI-
      C                                NATION OF MIDPOINTS OF THE MOHR'S CIRCLE.
      C----------------------------------------------------------------------
                DO 370 J=1,3
                   C(J    )=AB*A(J,J)-AC
                   C(J+ 5)=(HH(J,1)-HH(J,3))/V2
                   C(J+ 9)=(HH(J,1)+HH(J,3))/V2
                   C(J+14)=(HH(J,1)-HH(J,2))/V2
                   C(J+18)=(HH(J,1)+HH(J,2))/V2
                   C(J+23)=(HH(J,2)-HH(J,3))/V2
                   C(J+27)=(HH(J,2)+HH(J,3))/V2
        370     CONTINUE
                C( 4)=0.5*(A(1,1)-A(3,3))
                C( 9)=0.5*(A(1,1)+A(3,3))
                C(13)=0.5*(A(1,1)-A(2,2))
                C(18)=0.5*(A(1,1)+A(2,2))
                C(22)=0.5*(A(2,2)-A(3,3))
                C(27)=0.5*(A(2,2)+A(3,3))
                C( 5)=0.5*(C(1)-C(3))
                C(14)=0.5*(C(1)-C(2))
                C(23)=0.5*(C(2)-C(3))
          IF(C(13).GT.C(22))GO TO 385
                DO 380 I=1,9
                   C(I+30)=C(I+12)
                   C(I+12)=C(I+21)
                   C(I+21)=C(I+30)
        380     CONTINUE
        385 CONTINUE
      C
      C*********************************************************************
      C
      C     ITER BLOCK
            IF(ISKIP.GT.0)GO TO 699
            IF(EKEY.EQ.3)GO TO 603
      C     FLEXIBLE
            IF(IPOS.EQ.1)WRITE(6,906)C(1),AX,AY,DEPTH
            IF(IPOS.EQ.2)WRITE(6,907)B(3,3),AX,AY,DEPTH
            GO TO 611
      C     RIGID
        603 WRITE(6,908)A(1,1),AX,AY,DEPTH
        611 CONTINUE
            IF(EKEY.EQ.3)GO TO 620
            IF(EKEY2.GT.0)GO TO 630
      C     FLEXIBLE ALLOWABLE LOAD
            GO TO(601,604),IPOS
        601 IF(ABS(C(1)).LT.STRL2)GO TO 400
      C     ASPHALT CRITERIA CONTROLLING, SUBGRADE NOT NEEDED.
            IF(ITER.EQ.0) CALL NFRD(5,IDNM,1)
            ABSC1=C(1)
            IF(EKEY.EQ.1)WRITE(NOUT,915)ALOAD,PSI2,AA,BB,RN,ABSC1,STRL2
```

```
      IF(EKEY.EQ.2)WRITE(NOUT,916)ALOAD,PSI2,ES,XS,AS,BS,ABSC1,STRL2
      TAWGHT=(2.0*NLOAD*ALOAD)/0.95
      WRITE(NOUT,911)TAWGHT
      NPOS3=NLOAD
      CALL POST(X,Y,NPOS3,LAY,AXX,AYY,DEP,ETAA)
      ITER=1
      ISKIP=1
      GO TO 580
  604 IF(ABS(B(3,3)).LT.STRL)GO TO 680
      VSTR=B(3,3)
      IF(EKEY.EQ.1)WRITE(NOUT,909)ALOAD,PSI2,AA,BB,RN,VSTR,STRL
      IF(EKEY.EQ.2)WRITE(NOUT,910)ALOAD,PSI2,ES,XS,AS,BS,VSTR,STRL
      TAWGHT=(2.0*NLOAD*ALOAD)/0.95
      WRITE(NOUT,911)TAWGHT
      NPOS3=NLOAD
      CALL POST(X,Y,NPOS3,LAY,AXX,AYY,DEP,ETAA)
      ITER=1
      ISKIP=1
      GO TO 580
  620 CONTINUE
C     RIGID OVERLAY
      IF(EKEY2.NE.2)GO TO 640
      IF(ABS(A(1,1)).GE.STSL)GO TO 690
      ABSRS=ABS(A(1,1))
      WRITE(NOUT,912)DSM,ABSRS,STSL,THICK(1)
      NPOS3=NLOAD
      CALL POST(X,Y,NPOS3,LAY,AXX,AYY,DEP,ETAA)
      ITER=1
      ISKIP=1
      GO TO 580
  640 CONTINUE
C     RIGID ALLOWABLE LOAD
      IF(ABS(A(1,1)).LT.STSL)GO TO 680
      RSTS=A(1,1)
      WRITE(NOUT,913)ALOAD,PSI2,FS,DSM,RSTS,STSL
      TAWGHT=(2.0*NLOAD*ALOAD)/0.95
      WRITE(NOUT,911)TAWGHT
      NPOS3=NLOAD
      CALL POST(X,Y,NPOS3,LAY,AXX,AYY,DEP,ETAA)
      ITER=1
      ISKIP=1
      GO TO 580
  680 CONTINUE
C     INCREMENT LOAD
      DO 670 I7=1,NLOAD
      LDSTRS(I7)=ALOAD+ALIN
  670 LDSTRS(I7)=LDSTRS(I7)/CAREA
      ALOAD=ALOAD+ALIN
      LOAD=LDSTRS(1)
      PSI2=LOAD
      ITER=1
      GO TO 580
  630 CONTINUE
```

```
C      FLEXIBLE OVERLAY
       GO TO(631,632),IPOS
  631  CONE=C(1)
       GO TO 400
  632  IF(ABS(CONE).GE.STRL2 .OR. ABS(B(3,3)).GE.STRL)GO TO 690
       ABC1=ABS(CONE)
       ABSV12=ABS(B(3,3))
       WRITE(NOUT,914)DSM,ABC1,STRL2,ABSV12,STRL,THICK(1)
       NPOS3=NLOAD
       CALL POST(X,Y,NPOS3,LAY,AXX,AYY,DEP,ETAA)
       ITER=1
       ISKIP=1
       GO TO 580
  690  CONTINUE
C      INCREMENT THICKNESS
       THICK(1)=ATHICK+ATLIN
       ATHICK=ATHICK+ATLIN
       DO 666 IT=1,NPOS
  666  DEP(IT)=DEP(IT)+ATLIN
       ITER=1
       GO TO 580
  699  CONTINUE
C
C*************************************************************************
C
C------------------------------------------------------------------------
C                                   OUTPUT FOR PRINCIPAL STRESSES,ETC,MAXIMUM
C                                   SHEAR STRESSES,ETC AND STRAIN ENERGIES.
C------------------------------------------------------------------------
   390       WRITE(NOUT,9130)A(1,1),C(1),HH(1,1),HH(2,1),HH(3,1),
     +                       A(2,2),C(2),HH(1,2),HH(2,2),HH(3,2),
     +                       A(3,3),C(3),HH(1,3),HH(2,3),HH(3,3),
     +                       (C(I),I=4,30)
             BX = (A(1,1)*C(1)+A(2,2)*C(2)+A(3,3)*C(3))*0.5
             BY = 0.6666667*AB*(C(4)*C(4)+C(13)*C(13)+C(22)*C(22))
             WRITE(NOUT,9200) BX,BY
   400    CONTINUE
          GO TO 460
   410    WRITE(NOUT,9140) J
          GO TO 440
   420    WRITE(NOUT,9180) J
          GO TO 440
   430    WRITE (NOUT,9190)
C------------------------------------------------------------------------
C                                   FOR SYSTEMS FOR WHICH IT IS CLEAR THAT
C                                   MISTAKES OCCUR IN THE INPUTCARDS,THE
C                                   REQUEST AND POINT INPUT CARDS ARE SKIPPED.
C                                   PROGRAM PROCEEDS BY TAKING NEXT SYSTEM.
C------------------------------------------------------------------------
   440    READ (NIN,9150)
          READ(NIN,9030) NPOS
          DO 450 I=1,NPOS
   450    READ (NIN ,9150)
```

```
  460 CONTINUE
      WRITE(NOUT,9160)
      STOP
 900  FORMAT(1H1,//5(2H* ),
     1'F L E X I B L E   P A V E M E N T   A L L O W A B L E   ',
     2'L O A D ',5(2H* ),////60(1H*)/,
     3'STRL(SUBGRADE LIMITING STRAIN) = ',F15.8/,
     4'STRL2(ASPHALT LIMITING STRAIN) = ',F15.8/,
     5'ALOAD = ',F15.0/,'ALIN = ',F15.0/,
     660(1H*),////)
 901  FORMAT(1H1,//5(2H* ),
     A58HR I G I D   P A V E M E N T   A L L O W A B L E   L O A D ,
     B5(2H* ),////60(1H*)/,20X,'FS     = ',F6.0//,
     1                    20X,'DSM    = ',F6.0//,
     2                    20X,'ALOAD  = ',F6.0//,
     3                    20X,'ALIN   = ',F6.0//,
     4                    20X,'STSL   = ',F15.8,/60(1H*)////)
 902  FORMAT(1H1,//5(2H* ),19HO V E R L A Y   O V,
     141H E R   F L E X I B L E   P A V E M E N T ,5(2H* ))
 903  FORMAT(////60(1H*)/,20X,'DSM       = ',F6.0//,
     A                     20X,'SWL       = ',F12.0//,
     1                     20X,'ATHICK    = ',F6.2//,
     2                     20X,'ATLIN     = ',F6.2//,
     44X,'LIMITING SUBGRADE STRAIN = ',F15.8/,
     55X,'LIMITING ASPHALT STRAIN = ',F15.8/,60(1H*)////)
 904  FORMAT(1H1,//5(2H* ),23HO V E R L A Y   O V E R,
     131H   R I G I D   P A V E M E N T ,5(2H* ))
 905  FORMAT(////60(1H*)/,20X,'DSM       = ',F6.0//,
     A                     20X,'SWL       = ',F12.0//,
     B                     20X,'FS        = ',F6.0//,
     1                     20X,'ATHICK    = ',F6.2//,
     2                     20X,'ATLIN     = ',F6.2//,
     3                     20X,'STSL      = ',F15.8/,60(1H*)////)
 906  FORMAT('MAXIMUM NORMAL STRAIN = ',F15.8,3F15.2//)
 907  FORMAT('VERTICAL STRAIN = ',F15.8,3F15.2//)
 908  FORMAT('MAXIMUM NORMAL STRESS = ',F15.8,3F15.2//)
 909  FORMAT(//20X,'ALOAD = ',F10.0//,22X,'PSI = ',F10.2,//
     110X,'AA = ',F12.8,'   BB = ',F12.8,'   RN = ',F12.0,
     2//16(1H*),'VSTR = ',F12.8,'      STRL = ',F12.8,16(1H*)////)
 910  FORMAT(//20X,'ALOAD = ',0PF10.0//,22X,'PSI = ',0PF10.2,//
     1,10X,'ES = ',0PF12.0,3X,'XS = ',
     20PF8.0,7X,'AS = ',0PF12.6//,'BS = ',0PF12.6//,
     A16(1H*),'VSTR = ',0PF12.8,
     3'       STRL = ',0PF12.8,16(1H*)////)
 911  FORMAT(///20X,'TOTAL ALLOWABLE WEIGHT = ',F10.0///)
 912  FORMAT(1H1,60(1H*)/,60(1H*)/,20X,'DSM     = ',F6.0//,
     1                                20X,'ABSRS  = ',F15.8//,
     2                                20X,'STSL   = ',F15.8//,
     3                                20X,'FINAL THICKNESS = ',F8.2/,
     4                               60(1H*)/,60(1H*)////)
 913  FORMAT(1H1,60(1H*)/,60(1H*)/,20X,'ALOAD = ',F10.0//,
     1                                20X,'PSI    = ',F8.0//,
     2                                20X,'FS     = ',F6.0//,
```

```
     3                                 20X,'DSM    = ',F6.0//,
     4                                 20X,'RSTS   = ',F15.8//,
     5                                 20X,'STSL   = ',F15.8,
     6                                 /60(1H*)/,60(1H*)/////)
 914  FORMAT(1H1,60(1H*)/,60(1H*)/,20X,'DSM      = ',F6.0//,
     A                                 20X,'ABC1    = ',F15.8//,
     B                                 20X,'STRL2   = ',F15.8//,
     1                                 20X,'ABSV12 = ',F15.8//,
     2                                 20X,'STRL    = ',F15.8//,
     3                                 20X,'FINAL THICKNESS = ',F8.2/,
     4                                 60(1H*)/,60(1H*)/////)
 915  FORMAT(//20X,'ALOAD = ',F10.0//,22X,'PSI = ',F10.2,//
     110X,'AA = ',F12.8,'    BB = ',F12.8,'    RN = ',F12.0,
     2//16(1H*),'ABSC1 = ',F12.8,'      STRL2 = ',F12.8,16(1H*)/////)
 916  FORMAT(//20X,'ALOAD = ',0PF10.0//,22X,'PSI = ',0PF10.2,//
     1,10X,'ES = ',0PF12.0,3X,'XS = ',0PF8.0,7X,'AS = ',0PF12.6//,
     2'BS = ',0PF12.6//,16(1H*),'ABSC1 = ',0PF12.8,
     3'      STRL2 = ',0PF12.8,16(1H*)/////)
9000 FORMAT(1H1,17X,11('B'),5X,'III',5X,11('S'),6X,9('A'),5X,11('R')/
     +                  18X,12('B'),4X,'III',4X,12('S'),5X,11('A'),4X,12('R')/
     +18X,'BB',8X,'BBB    III    SSS',14X,'AAA',7X,'AAA    RR',8X,'RRR'/
     +18X,'BB',9X,'BB    III    SS',15X,'AA',9X,'AA    RR',9X,'RR'/
     +18X,'BB',7X,'BBB     III     SSS',14X,'AA',9X,'AA    RR',8X,'RR'/
     +18X,11('B'),5X,'III',4X,11('S'),5X,13('A'),3X,12('R')/
     +18X,11('B'),5X,'III',5X,11('S'),4X,13('A'),3X,11('R')/
     +18X,'BB',7X,'BBB     III',14X,'SSS    AA',9X,'AA    RR',5X,'RR'/
     +18X,'BB',8X,'BB     III',15X,'SS    AA',9X,'AA    RR',6X,'RR'/
     +18X,'BB',7X,'BBB     III',14X,'SSS    AA',9X,'AA    RR',7X,'RR'/
     +18X,12('B'),4X,'III',3X,13('S'),4X,'AA',9X,'AA    RR',8X,'RR'/
     +18X,11('B'),5X,'III',3X,12('S'),5X,'AA',9X,'AA    RR',9X,'RR'////
     +75X,'THIS ''BISAR'' PROGRAM HAS BEEN OBTAINED FROM'/39X,'SHELL RES
     +EARCH B.V.'/89X,'FOR THE SOLE USE OF'//76X,
     +'SHELL OIL COMPANY'/76X,'HOUSTON, TEXAS'
     +                                     //76X,'ALL RIGHTS ARE RESERVED.
     + USE OF THIS PROGRAM'/76X,'BY UNAUTHORIZED PERSONS IS PROHIBITED')
9010 FORMAT(15A4)
9020 FORMAT(1H1,25(/),15X,15A4)
9030 FORMAT(I2,I3,I1)
9040 FORMAT(' NOTE THAT ',E12.6,' AND ',E12.6,' WILL BE CONSIDERED TO B
     +E LOADS IN STRESS UNITS')
9050 FORMAT(4E12.6)
9060 FORMAT(26A3,A2)
9070 FORMAT(' NOTE THAT INCORRECT SPELLING HAS NOT STOPPED THE EVALUATI
     +ON OF STRESS',4X,A3)
9080 FORMAT(I2,4E12.6)
9090 FORMAT(1H1,///52X,'POSITION NUMBER ',I2//54X,'LAYER NUMBER  ',I2//
     +55X,'COORDINATES'//46X,'X',11X,'Y',11X,'Z'/40X,3E12.4)
9100 FORMAT(/21X,'DISTANCE TO LOAD-AXIS(',I2,')',34X,'THETA'/25X,E12.4,
     +41X,E12.4/)
9110 FORMAT(//,30X,'THIS POSITION HAS BEEN OMITTED SINCE THE LAYER NUMB
     +ER IS INCORRECT')
9120 FORMAT(/30X,'XX',10X,'YY',10X,'ZZ',10X,'YZ',10X,'XZ',10X,'XY',10X,
     +'UX',10X,'UY',10X,'UZ')
```

```
 9130 FORMAT(/' P R I N C I P A L   V A L U E S   A N D   D I R E C T I O N00017750
     + S   O F   T O T A L   S T R E S S E S   A N D   S T R A I N S'/15X,'NO00017760
     +RMAL',9X,'NORMAL',9X,'SHEAR',10X,'SHEAR',13X,'X',14X,'Y',14X,'Z'/100017770
     +5X,'STRESS',9X,'STRAIN',9X,'STRESS',9X,'STRAIN',9X,'COMPONENT',6X,00017780
     +'COMPONENT',6X,'COMPONENT'/' MAXIMUM',2E15.3,30X,3F15.3/' MINIMAX'00017790
     +,2E15.3,30X,3F15.3/' MINIMUM',2E15.3,30X,3F15.3/' MAXIMUM',30X,2E100017800
     +5.3,3F15.3/8X,E15.3,45X,3F15.3/' MINIMAX',30X,2E15.3,3F15.3/8X,   00017810
     +E15.3,45X,3F15.3/' MINIMUM',30X,2E15.3,3F15.3/8X,E15.3,45X,3F15.3)00017820
 9140 FORMAT(' THE PROBLEM CANNOT BE SOLVED,NU(',I2,') EQUALS ONE')     00017830
 9150 FORMAT(/)                                                          00017840
 9160 FORMAT(1H1)                                                        00017850
 9170 FORMAT(A4,6X,6F10.0)                                               00017860
 9180 FORMAT(' THE PROBLEM CANNOT BE SOLVED,E(',I2,') EQUALS ZERO')      00017870
 9190 FORMAT(' SYSTEM SKIPPED NO LOADS')                                 00017880
 9200 FORMAT(1H0,13X,' STRAIN ENERGY',E11.4/' STRAIN ENERGY OF DISTORTIO00017890
     +N',E11.4)                                                          00017900
      END                                                                00017910
      SUBROUTINE SYSTEM(ISYS,E,NU,THICK,AK,NLAYS,M,NLOAD,LDSTRS,HOSTR,   00017920
     1ALK,RADIUS,X,Y,PSI,ISMO,IRED)                                      00017930
C-----------------------------------------------------------------------00017940
C                                 THIS SUBROUTINE OUTPUTS ALL PHYSICAL DATA 00017950
C                                 OF THE MULTI-LAYERED SYSTEM AND ALL DATA  00017960
C                                 ON CONFIGURATION AND MAGNITUDE OF THE     00017970
C                                 LOADS.                                    00017980
C-----------------------------------------------------------------------00017990
      INTEGER ROUGH(2),SMOOTH(2),ISMTH(2)                                00018000
      REAL E(10),NU(10),THICK(9),AK(9),ALK(9),LDSTRS(10),HOSTR(10),      00018010
     1RADIUS(10),X(10),Y(10),PSI(10)                                     00018020
      COMMON/TAPE/NOUT                                                   00018030
      DATA ROUGH,SMOOTH/'ROU','GH ','SMO','OTH'/                         00018040
      WRITE(NOUT,1001) ISYS                                              00018050
      IF(IRED.EQ.0) WRITE(NOUT,1002)                                     00018060
      IF(IRED.NE.0) WRITE(NOUT,1007)                                     00018070
      IF(NLAYS.EQ.1) GO TO 40                                            00018080
      DO 30 I=1,M                                                        00018090
         IF(ISMO.EQ.1) GO TO 10                                          00018100
         ISMTH(1) = ROUGH(1)                                             00018110
         ISMTH(2) = ROUGH(2)                                             00018120
         IF(ALK(I).LT.100.0) GO TO 20                                    00018130
   10    ISMTH(1) = SMOOTH(1)                                            00018140
         ISMTH(2) = SMOOTH(2)                                            00018150
   20    WRITE(NOUT,1003) I,ISMTH(1),ISMTH(2),E(I),NU(I),THICK(I),AK(I) 00018160
   30 CONTINUE                                                           00018170
   40 WRITE(NOUT,1004) NLAYS,E(NLAYS),NU(NLAYS)                          00018180
      WRITE(NOUT,1005)                                                   00018190
      DO 50 I = 1,NLOAD                                                  00018200
   50 WRITE(NOUT,1006) I,LDSTRS(I),HOSTR(I),RADIUS(I),X(I),Y(I),PSI(I)  00018210
 1001 FORMAT(1H1,10(/),52X,'SYSTEM NUMBER',3X,I2)                        00018220
 1002 FORMAT(5(/),8X,'LAYER',4X,'CALCULATION',2X,'YOUNG''S',4X,'POISSON'00018230
     1'S',3X,'THICKNESS',3X,'INTERFACE'/8X,'NUMBER',3X,'METHOD',7X,'MODU00018240
     2LUS',4X,'RATIO',18X,'SPRINGCOMPL'/)                                00018250
 1003 FORMAT(10X,I2,5X,2A3,3X,4E12.4)                                    00018260
 1004 FORMAT(10X,I2,14X,2E12.4)                                          00018270
```

```
 1005 FORMAT(///8X,'LOAD',5X,'NORMAL',7X,'SHEAR',5X,'RADIUS OF',7X,'LOAD
     1 - POSITION',6X,'SHEAR'/8X,'NUMBER',3X,'STRESS',7X,'STRESS',4X,
     2'LOADED AREA',6X,'X',11X,'Y',7X,'DIRECTION'/)
 1006 FORMAT(10X,I2,2X,6E12.4)
 1007 FORMAT(5(/),8X,'LAYER',4X,'CALCULATION',2X,'YOUNG''S',4X,'POISSON'
     1'S',3X,'THICKNESS',3X,' REDUCED' /8X,'NUMBER',3X,'METHOD',7X,'MODU
     2LUS',4X,'RATIO',18X,'SPRINGCOMPL'/)
      RETURN
      END
      SUBROUTINE MACON1(ISMO,ALK,NSYS)
C-----------------------------------------------------------------------
C                                THIS SUBROUTINE CALCULATES CONSTANTS USED
C                                IN SUBROUTINE MATRIX TO BUILD UP VARIOUS
C                                MATRICES.
C                                THE CONSTANTS ARE STORED IN
C                                COMMON/INDATA/.
C                                NUMERICAL STABILITY OF SOLUTIONPROCEDURE
C                                FOR THE SYSTEM IS TESTED BY CALLING IN#
C                                         MA2CON,
C                                         MATRIX
C                                WHEN INSTABILITY HAS TO BE EXPECTED THE
C                                SMOOTH CALCULATION PROCEDURE IS CHOSEN BY
C                                TAKING ISMO = 1 AND NSYS IS SET EQUAL 1 .
C-----------------------------------------------------------------------
      REAL K1,K2,K3(10),K4(10),K5,K6,NU,II,LOAD,ACCUR(3),ALK(9)
      COMMON/ASDT/LAYER,NLAYS,M,R,Z,NU(10),ACCUR,LOAD,HOSTRS,NZEROS,H(9)
     1,K5(10),E(10),AL(9),THICK(9),RADIUS(10)
      COMMON/INDATA/XMAX,  A1(9),B1(9),C1(9),D(9),EE(9),F(9),G(9),H1(9),
     1II(9),K1(9),K2(9),K6(10),BE(9),BU(9),BUU(9),BMU(9),B2U(9),B2UU(9),
     2J2(9),J1,T2(10),SS(2,10),G012(9),G021(9),G022(9),G122(9),
     3H012(9),H022(9),H122(9),D012(9),D022(9),C011(9),C012(9),E012(9),
     4F012(9),F112(9),F022(9),CC(4,2,9),DD(2,2,9),FF(2,2,9),GG(2,2,10),
     5HH(2,2,10),RR(4,2,10),DD2(9),G20(9),G21(9),H20(9),H021(9),GG2(10),
     6HH2(10),Q011(9),Q111(9),Q012(9),Q112(9),Q212(9),Q022(9),Q122(9),
     7QF0(9),QF1(9),Z011,Z111,Z211,Z012,Z112,Z212,Z312,Z021,Z121,Z022,
     8Z122,Z222,K4
      COMMON/TAPE/NOUT
      NSYS = 0
      IF(NLAYS.EQ.1) GO TO 10
      GG(1,1,1) = -1.0
      GG(2,1,1) = 1.0
      GG(1,2,1) = 1.0-2.0*NU(1)
      GG(2,2,1) = 2.0*NU(1)
      HH(1,1,1) = 1.0
      HH(1,2,1) = GG(1,2,1)
      HH(2,1,1) = 1.0
      HH(2,2,1) = -GG(2,2,1)
      RR(1,1,NLAYS) = 0.0
      RR(1,2,NLAYS) = 0.0
      RR(2,1,NLAYS) = 0.0
      RR(2,2,NLAYS) = 0.0
      RR(3,1,NLAYS) = 1.0
      RR(3,2,NLAYS) = 0.0
```

```
      RR(4,1,NLAYS) = 0.0
      RR(4,2,NLAYS) = 1.0
      SS(1,NLAYS) = 0.0
      SS(2,NLAYS) = 1.0
      GG2(1) = 1.0
      HH2(1) =-1.0
   10 K5(1)=1.0-2.0*NU(1)
      IF(NLAYS.EQ.1) GO TO 70
      K = 0
      K6(1) = 4.0*(1.0-NU(1))
      DO 30 J=1,M
         K1(J)=(1.0+NU(J+1))*E(J)/((1.0+NU(J))*E(J+1))
         K2(J)=1.0-K1(J)
         K3(J)=NU(J+1)-NU(J)*K1(J)
         K4(J)=8.0*NU(J)*NU(J+1)
         K5(J)=1.0-2.0*NU(J)
         K6(J+1) = 4.0*(1.0-NU(J+1))
         A1(J)=     K6(J)-K2(J)
         B1(J)=     K2(J)+K1(J)*K6(J+1)
         C1(J)=2.0*K2(J)
         D (J)=     K2(J)*(1.0-4.0*NU(J))
         EE(J)=     K2(J)*(1.0+K4(J))-6.0*K3(J)
         F (J)=     A1(J)-B1(J)
         G (J)=     K2(J)*(1.0-K4(J))+2.0*K3(J)
         H1(J)=4.0*K2(J)*(NU(J+1)-NU(J))
         II(J)=     D(J)-H1(J)
   30 CONTINUE
      K5(M+1)=1.0-2.0*NU(M+1)
      IF(ISMO.EQ.1) GO TO 70
      DO 40 I = 1,M
         IF(ALK(I).LT.100.0) GO TO 50
   40 CONTINUE
      GO TO 70
C-----------------------------------------------------------------------
C                                 CALCULATION OF CONSTANTS ONLY NEEDED IN
C                                 MATRIX FOR STABILITY TEST.
C-----------------------------------------------------------------------
   50 TMIN = 1.0E+10
      NTELL = 2
      DUMMY = 0.0
      LAYER = NLAYS
      T2(NLAYS) = 0.0
      DO 60 K = 1,M
         IF(THICK(K).LT.TMIN) TMIN = THICK(K)
         DUMMY = DUMMY+THICK(K)
         T2(K) = 2.0*THICK(K)/RADIUS(1)
         H(K) = DUMMY/RADIUS(1)
   60 CONTINUE
      CALL MA2CON(TMIN,1,ISMO,ALK)
      TX = 6.6*RADIUS(1)/TMIN
      XMAX = TX+1.0
C-----------------------------------------------------------------------
C                                 TEST ON NUMERICAL STABILITY OF THE SOLU-
```

```
C                                  TION-PROCEDURE TO BE FOLLOWED FOR THIS
C                                  SYSTEM BY CALLING IN THE MATRIX SUBROUTI-
C                                  NE WITH NTELL = 2 .
C                                  AFTER TEST THE SMOOTH OR ROUGH CALCULATI-
C                                  ON PROCEDURE IS CHOSEN.
C                                  TEST IS ONLY NECESSARY IF NOT DIRECTLY
C                                  THE SMOOTH CALCULATION PROCEDURE HAS BEEN
C                                  CHOSEN BY ISMO=1.
C-----------------------------------------------------------------------
      CALL MATRIX(TX,1,NTELL)
      IF(NTELL.EQ.2) GO TO 70
      ISMO = 1
      NSYS = 1
      WRITE(NOUT,1001)
   70 RETURN
 1001 FORMAT(' THE MORE STABLE SMOOTH CALCULATION PROCEDURE HAS BEEN CHO
     1SEN.')
      END
      SUBROUTINE CONSYS(AID,NZEP,NZEQ,N,L)
C-----------------------------------------------------------------------
C                                  THIS SUBROUTINE DETERMINES FOR EACH SYS-
C                                  TEM THE CYLINDRICAL COMPONENTS NEEDED FOR
C                                  COMPUTATION OF THE REQUIRED CARTESIAN
C                                  COMPONENTS OF STRESSES,STRAINS AND DISPLA-
C                                  CEMENT. GIVEN THIS SET OF COMPONENTS A
C                                  FURTHER SELECTION IS PERFORMED ON THE
C                                  COMPONENTS THAT CAN BE COMPUTED WITH THE
C                                  INTEGRALS.
C                                  CONSYS CALLS IN SUBROUTINE LOGSET
C-----------------------------------------------------------------------
      LOGICAL AID(27),NZEP,NZEQ,EPS(5),N,L
      INTEGER JARG(6,14)
      DATA JARG/
     1 4, 5, 7,18,19,21,       8, 9,20,22, 0, 0,      10,11,13,23,24,26,
     214,15,25,27, 0, 0,       1, 2,16,17, 0, 0,       5,10,12, 0, 0, 0,
     3 4,10,12, 0, 0, 0,       4, 5,10, 0, 0, 0,       4, 6,12, 0, 0, 0,
     4 4, 5,12, 0, 0, 0,      16,17, 0, 0, 0, 0,      18,19,21,23,24,26,
     523,24,26, 0, 0, 0,      20,22,25,27, 0, 0/
      EPS(1) = AID(18).OR.AID(19).OR.AID(21)
      EPS(2) = AID(20).OR.AID(22)
      EPS(3) = AID(23).OR.AID(24).OR.AID(26)
      EPS(4) = AID(25).OR.AID(27)
      EPS(5) = AID(16).OR.AID(17)
      DO 10 I = 1,5
         IF(.NOT.EPS(I)) GO TO 10
         CALL LOGSET(JARG(1,I),AID)
   10 CONTINUE
      IF(AID(10).AND..NOT.NZEQ) AID( 4)=.TRUE.
      IF(AID( 6).AND..NOT.NZEP) AID(12)=.TRUE.
      IF(AID( 4)) CALL LOGSET(JARG(1, 6),AID)
      IF(AID(10)) AID(11)=.TRUE.
      IF(AID( 5)) AID(11)=.TRUE.
      IF(AID(12)) AID( 6)=.TRUE.
```

```
      IF(AID( 8)) AID(14)=.TRUE.
      IF(AID(14)) AID( 8)=.TRUE.
      IF(AID( 5).AND.AID( 6)) CALL LOGSET(JARG(1, 7),AID)
      IF(AID(11).AND.AID(12)) CALL LOGSET(JARG(1, 8),AID)
      IF(.NOT.NZEQ) GO TO 20
      IF(AID( 7)) AID(13)=.TRUE.
      IF(AID(13)) AID( 7)=.TRUE.
      IF(AID( 9)) AID(15)=.TRUE.
      IF(AID(15)) AID( 9)=.TRUE.
      IF(AID( 5).AND.AID(10)) CALL LOGSET(JARG(1, 9),AID)
      IF(AID( 6).AND.AID(10)) CALL LOGSET(JARG(1,10),AID)
      IF(AID( 1).AND.AID( 2)) CALL LOGSET(JARG(1,11),AID)
      IF(AID( 4).AND.AID( 7)) CALL LOGSET(JARG(1,12),AID)
      IF(AID( 7).AND.AID(10)) CALL LOGSET(JARG(1,13),AID)
      IF(AID( 8).AND.AID( 9)) CALL LOGSET(JARG(1,14),AID)
      GO TO 30
   20 IF(AID( 1)) CALL LOGSET(JARG(1,11),AID)
      IF(AID( 4)) CALL LOGSET(JARG(1,12),AID)
      IF(AID( 8)) CALL LOGSET(JARG(1,14),AID)
   30 N = .FALSE.
      L = .TRUE.
      IF(AID(3).OR.AID(6).OR.AID(12).OR.AID(16).OR.AID(17)) N=.TRUE.
      DO 50 I = 18,27
         IF(AID(I)) GO TO 40
         L = .FALSE.
         GO TO 50
   40 N = .TRUE.
   50 CONTINUE
      RETURN
      END
      SUBROUTINE LOGSET(I,LOG)
C-----------------------------------------------------------------------
C                                THIS SUBROUTINE,CALLED IN BY CONSYS AND
C                                CONPNT,SETS THE LOGICAL VARIABLES LOG(K)
C                                TRUE FOR THE K-VALUES,STORED IN THE ARGU-
C                                MENT I.
C-----------------------------------------------------------------------
      LOGICAL LOG(1)
      INTEGER I(1)
      DO 10 L=1,6
         IF(I(L).EQ.0)         GO TO 20
         K=I(L)
         LOG(K)=.TRUE.
   10 CONTINUE
   20 RETURN
      END
      SUBROUTINE MA2CON(TMIN,I,ISMO,ALK)
C-----------------------------------------------------------------------
C                                THIS SUBROUTINE CALCULATES CONSTANTS USED
C                                IN SUBROUTINE MATRIX TO BUILD UP VARIOUS
C                                MATRICES. THESE CONSTANTS ALL DEPENDENT
C                                ON ALK(J) AND / OR RADIUS(I), ARE STORED
C                                IN COMMON/INDATA/.
```

```
C-----------------------------------------------------------------------
      REAL K1,K2,K4(10),K5,K6,K11,K12,NU,II,LOAD,ACCUR(3),ALK(9)
      COMMON/ASDT/LAYER,NLAYS,M,R,Z,NU(10),ACCUR,LOAD,HOSTRS,NZEROS,H(9)
     1,K5(10),E(10),AL(9),THICK(9),RADIUS(10)
      COMMON/INDATA/XMAX,  A1(9),B1(9),C1(9),D(9),EE(9),F(9),G(9),H1(9),
     1II(9),K1(9),K2(9),K6(10),BE(9),BU(9),BUU(9),BMU(9),B2U(9),B2UU(9),
     2J2(9),J1,T2(10),SS(2,10),G012(9),G021(9),G022(9),G122(9),
     3H012(9),H022(9),H122(9),D012(9),D022(9),C011(9),C012(9),E012(9),
     4F012(9),F112(9),F022(9),CC(4,2,9),DD(2,2,9),FF(2,2,9),GG(2,2,10),
     5HH(2,2,10),RR(4,2,10),DD2(9),G20(9),G21(9),H20(9),H021(9),GG2(10),
     6HH2(10),Q011(9),Q111(9),Q012(9),Q112(9),Q212(9),Q022(9),Q122(9),
     7QF0(9),QF1(9),Z011,Z111,Z211,Z012,Z112,Z212,Z312,Z021,Z121,Z022,
     8Z122,Z222,K4
      XMAX =6.5*RADIUS(I)/TMIN
      K = 0
      DO 30 J = 1,M
         AL(J) = ALK(J)/(RADIUS(I)+ALK(J))
         K12 = 1.0-AL(J)
         Q011(J) = K12*B1(J)
         Q111(J) = 2.0*AL(J)*NU(J+1)
         Q012(J) =-K12*EE(J)
         Q022(J) = K12*A1(J)
         Q122(J) = AL(J)*K5(J)
         QF0(J) = K12*A1(J)*B1(J)
         QF1(J) = 2.0*AL(J)*(1.0-NU(J)+(1.0-NU(J+1))*K1(J))
         IF(ISMO.EQ.1) GO TO 20
         IF(ALK(J).GE.100.0) GO TO 20
         BE(J) = -AL(J)/(1.0-AL(J))
         BU(J) = BE(J)*2.0*NU(J+1)
         BUU(J) = BU(J)*K5(J)
         BMU(J) = BE(J)*K5(J)
         B2U(J) = BE(J)*(K5(J)-2.0*NU(J+1))
         B2UU(J) = BE(J)*(K5(J)+2.0*NU(J+1))
         GO TO 30
   20    K11 = 2.0*(NU(J)-NU(J+1))
         K = K+1
         J2(K) = J
         GG(1,1,K+1) = K2(J)
         G012(K) = K11-K2(J)*(2.0-4.0*NU(J+1))
         G021(K) =-K12*K2(J)
         G022(K) = (K11-K2(J))*K12
         G122(K) =-2.0*NU(J+1)*AL(J)
         HH(1,1,K+1) =-3.0+4.0*NU(J)-K1(J)
         H012(K) =-2.0+2.0*NU(J)+6.0*NU(J+1)-K4(J)-(2.0-4.0*NU(J+1))*
     1            K1(J)
         H021(K) = HH(1,1,K+1)*K12
         H022(K) =-K12*(1.0-2.0*NU(J)-6.0*NU(J+1)+K4(J)-K1(J))
         H122(K) = 2.0*AL(J)*NU(J+1)
         DD(1,1,K) =-K6(J)
         D012(K) =-K6(J)*K5(J)
         DD(2,1,K) =-K12*K6(J)
         D022(K) =-NU(J)*2.0*DD(2,1,K)
         C011(K) =-1.0+4.0*NU(J)
```

```
          C012(K) = 4.0*NU(J)*K5(J)
          CC(2,1,K) =-2.0
          E012(K) =-K11
          F012(K) = K4(J)-2.0*(NU(J)+NU(J+1))
          F112(K) = 2.0*E012(K)
          F022(K) = 1.0-4.0*NU(J+1)
          FF(2,1,K) = 2.0
          DD2(K) = 2.0*K12/K1(J)
          G20(K) = K12*(1.0-1.0/K1(J))
          G21(K) = AL(J)*0.5/K1(J)
          H20(K) = K12*(1.0+1.0/K1(J))
   30 CONTINUE
      J1 = K
      DUMMY=0.0
      T2(NLAYS) = 0.0
      DO 40 K = 1,M
          DUMMY=DUMMY+THICK(K)
          T2(K)=2.0*THICK(K)/RADIUS(I)
           H(K)=DUMMY/RADIUS(I)
          K12 = 1.0-AL(K)
          Q112(K) = -K12*F(K)*H(K)+Q111(K)*K5(K)
          Q212(K) = AL(K)*H(K)*(2.0*NU(K+1)-K5(K))
   40 CONTINUE
      IF(LAYER.EQ.NLAYS) GO TO 50
C-----------------------------------------------------------------------
C                                THESE CONSTANTS ARE USED FOR THE ASYMPTO-
C                                TIC EVALUATION OF THE CHARACTERISTIC
C                                FUNCTIONS IN MATRIX.
C-----------------------------------------------------------------------
      J = LAYER
      RK1 = 2.0*NU(J+1)*C1(J)
      RK2 = 2.0*NU(J+1)*A1(J)
      RK3 = 2.0*NU(J+1)*D(J)
      Z021 = Q011(J)*C1(J)
      RK4 = Z021*H(J)
      K12 = 1.0-AL(J)
      Z011 = Q011(J)*D(J)
      Z111 = AL(J)*(RK3-G(J)-K5(J)*B1(J))-RK4
      Z211 = AL(J)*H(J)*(B1(J)-II(J)-RK1)
      Z012 = -K12*(D(J)*EE(J)+A1(J)*G(J))
      Z112 = Q122(J)*(RK2-G(J)+RK3+EE(J))+K12*H(J)*(A1(J)*
     1        H1(J)+C1(J)*EE(J)-D(J)*F(J))
      Z212 = -AL(J)*H(J)*(K5(J)*(RK1+B1(J)+II(J))-RK3+EE(J)+
     1         RK2+G(J))-RK4*H(J)
      Z312 = AL(J)*H(J)*H(J)*(B1(J)-RK1-II(J))
      Z121 = AL(J)*(RK1-B1(J)+II(J))
      Z022 = K12*(A1(J)*II(J)-EE(J)*C1(J))
      Z122 = AL(J)*(RK2+EE(J)+K5(J)*(RK1+II(J)))+RK4
      Z222 = Z121*H(J)
   50 RETURN
      END
      SUBROUTINE CONPNT(R,HOSTRS,LOAD,Z,N2,L2)
C-----------------------------------------------------------------------
```

```
C                                   THIS SUBROUTINE DETERMINES FOR EACH POINT-
C                                   LOAD CONFIGURATION SEPARATELY THE
C                                   INTEGRALS NEEDED FOR COMPUTATION OF THE
C                                   DESIRED COMPONENTS OF STRESS,ETC.
C                                   FOR POINTS AT THE RIM OF THE LOAD SOME
C                                   COMPONENTS CANNOT BE CALCULATED BECAUSE
C                                   OF SINGULAR BEHAVIOUR, A MESSAGE IS
C                                   PRINTED.
C-----------------------------------------------------------------------
      LOGICAL STRESS,EPS,RLOW,N2,L2
      REAL LOAD
      INTEGER IARG(6,12),KARG(6,4),JJ(12,15)
      COMMON/STRDTA/STRESS(27),EPS(17),RLOW,ST,CT,L,ACC
      COMMON/TAPE/NOUT
      DATA IARG/
     1 7,12,17, 0, 0, 0,       12,14,17, 0, 0, 0,      10,11, 0, 0, 0, 0,
     2 7, 8, 9,12,14,17,        7, 9,12,14,17, 0,       8, 9, 0, 0, 0, 0,
     3 7,12,14,15,17, 0,        6,10,16, 0, 0, 0,      10,13,16, 0, 0, 0,
     4 7, 8,12,14,17, 0,        7,12,14,17, 0, 0,       8, 9, 0, 0, 0, 0/
      DATA KARG/
     1 1, 2, 4, 0, 0, 0,        2, 4, 0, 0, 0, 0,       1, 2, 0, 0, 0, 0,
     2 4, 5,10,11,12, 0/
      DATA JJ/
     1 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     2 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     3 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
     4 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
     5 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
     6 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0,
     7 1, 0, 1,-1,-1, 0, 0, 0, 0,-1, 1,-1,
     8 0, 1, 0, 0, 0, 0,-1, 0, 0, 0, 0, 0,
     9 1, 1, 1,-1,-1, 0,-1, 0, 0,-1, 1,-1,
     T 1, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1,
     1 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
     2 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1,
     3 1, 0, 1, 1, 1, 1, 0, 1, 0, 1, 1, 1,
     4 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
     5 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1/
      NERR = 0
      DO 10 I = 1,17
         EPS(I) = .FALSE.
   10 CONTINUE
      J=3
      IF(ABS(ST).LT.ACC)         J=1
      IF(ABS(CT).LT.ACC)         J=2
      IF(HOSTRS.LT.ACC)          GO TO 20
      IF(RLOW)                   GO TO 30
      IF(Z.LT.ACC)               GO TO 40
      I=J+14
      GO TO 50
   20 I=2
      IF(RLOW)                   I=1
      GO TO 50
```

```
   30 I=J+5
      IF(Z.LT.ACC)              I=I-3
      GO TO 50
   40 I=J+11
      IF(ABS(R-1.0).LT.ACC)     I=I-3
   50 IF(STRESS( 4).OR.STRESS(10))CALL LOGSET(KARG(1,1),EPS)
      IF(STRESS( 5))             CALL LOGSET(KARG(1,2),EPS)
      IF(STRESS( 3))             EPS( 3)=.TRUE.
      IF(STRESS(11))             EPS( 4)=.TRUE.
      IF(STRESS(12))             CALL LOGSET(KARG(1,3),EPS)
      IF(STRESS( 6).AND.(Z.GT.ACC)) EPS( 1)=.TRUE.
      IF(.NOT.STRESS( 8))        GO TO 60
      IF(Z.LT.ACC)               GO TO 60
      IF(R.GT.ACC)               EPS(5)=.TRUE.
   60 IF(I.LT.3) GO TO 180
      DO 90 J = 1,12
         IF (.NOT.STRESS(J))     GO TO 90
         IF(JJ(J,I-2))           70,90,80
   70    NERR = 1
         STRESS(J) = .FALSE.
         L2 = .FALSE.
         GO TO 90
   80    CALL LOGSET(IARG(1,J),EPS)
   90 CONTINUE
      IF(NERR) 160,160,100
  100 IF(I-10) 110,130,120
  110 WRITE(NOUT,9000)
      GO TO 140
  120 WRITE(NOUT,9020)
      STRESS(13) = .FALSE.
      GO TO 140
  130 WRITE(NOUT,9010)
      STRESS(13) = .FALSE.
      IF(STRESS(12)) GO TO 150
  140 IF(STRESS(3).OR.STRESS(6)) GO TO 150
      IF(STRESS(16)) GO TO 150
      IF(STRESS(17)) GO TO 150
      IF(STRESS(20)) GO TO 150
      IF(STRESS(22)) GO TO 150
      IF(STRESS(25)) GO TO 150
      IF(STRESS(27)) GO TO 150
      N2 = .FALSE.
  150 STRESS(18) = .FALSE.
      STRESS(19) = .FALSE.
      STRESS(21) = .FALSE.
      STRESS(23) = .FALSE.
      STRESS(24) = .FALSE.
      STRESS(26) = .FALSE.
  160 IF(LOAD.GT.ACC) GO TO 180
      DO 170 J = 1,5
  170    EPS(J) = .FALSE.
  180 RETURN
 9000 FORMAT(' AT THIS POINT SRR,STT,ERR AND EZZ HAVE A LOGARITHMIC SING
```

```
     1ULARITY')
 9010 FORMAT(' AT THIS POINT SRT AND ERT HAVE A LOGARITHMIC SINGULARITY'
     1)
 9020 FORMAT(' AT THIS POINT SRR,STT,SRT,ERR,EZZ AND ERT HAVE A LOGARITH
     1MIC SINGULARITY')
      END
      SUBROUTINE GENDAT(N,NZEROS,R,ACC)
C-----------------------------------------------------------------------
C                                THIS SUBROUTINE GIVES THE ZEROS OF THE
C                                PRODUCTS J0(XR)*J1(X) AND J1(XR)*J1(X) IN
C                                THE RIGHT ORDER. THE SUBSEQUENT ZEROS ARE
C                                STORED IN ZEROS FOR USING THEM IN INGRAL.
C                                THE ZEROS OF J0 AND J1 ARE STORED AS
C                                BZEROS IN THE BLOCK DATA.
C-----------------------------------------------------------------------
      COMMON/GEDATA/BZEROS(149,2),ZEROS(298)
      IF(R.LT.ACC.OR.ABS(R-1.0).LT.ACC) GO TO 40
      I=1
      J=1
      DO 20 K=1,298
         IF(I.GT.149) GO TO 30
         IF(J.GT.149) GO TO 30
         IF(BZEROS(I,2).LT.BZEROS(J,N+1)/R) GO TO 10
         ZEROS(K) = BZEROS(J,N+1)/R
         J=J+1
         GO TO 20
   10    ZEROS(K)=BZEROS(I,2)
         I=I+1
   20 CONTINUE
   30 NZEROS = K-1
      RETURN
   40 IF(R.GT.ACC) GO TO 70
   50 DO 60 I=1,149
         ZEROS(I)=BZEROS(I,2)
   60 CONTINUE
      NZEROS=149
      RETURN
   70 IF(N.EQ.1) GO TO 50
      DO 80 K=1,149
         ZEROS(2*K-1)=BZEROS(K,1)
         ZEROS(2*K  )=BZEROS(K,2)
   80 CONTINUE
      NZEROS=298
      RETURN
      END
      SUBROUTINE ASYMPT(R,ACC)
C-----------------------------------------------------------------------
C                                THIS SUBROUTINE ORGANIZES THE COMPUTATION
C                                OF THE ASYMPTOTIC PART OF THE INTEGRALS
C                                AS USED FOR THE TOP-LAYER ONLY.
C                                ASYMPT CALLS IN SUBROUTINE   ASS
C                                ASYMPT CALLS IN FUNCTIONS    FLLE
C                                                             FLLK
```

```
      C                                                 FLMBDA        00023050
      C-----------------------------------------------------------------00023060
            DOUBLE PRECISION DR,KACC2,C,ELLE,ELLK,FLLE,FLLK               00023070
            COMMON/CONST/C,ELLE,ELLK,ALMBDA                                00023080
            IF(R.LT.ACC) GO TO 10                                          00023090
            DR=DBLE(R)                                                     00023100
            KACC2=((1.0D0-DR)*(1.0D0-DR)+C*C)/((1.0D0+DR)*(1.0D0+DR)+C*C)  00023110
            ELLK=FLLK(KACC2)                                               00023120
            ELLE=FLLE(KACC2)                                               00023130
            ALMBDA = FLMBDA(DR,C,ELLK,ELLE,KACC2)                          00023140
         10 CALL ASS(ACC,R)                                                00023150
            RETURN                                                         00023160
            END                                                            00023170
            DOUBLE PRECISION FUNCTION FLLK(KACC2)                          00023180
      C-----------------------------------------------------------------00023190
      C                              THIS FUNCTIONSUBROUTINE EVALUATES THE   00023200
      C                              COMPLETE ELLIPTIC INTEGRAL OF THE FIRST 00023210
      C                              KIND FROM A SERIES-EXPANSION ACCORDING TO 00023220
      C                              BYRD AND FRIEDMAN,HANDBOOK OF ELLIPTIC  00023230
      C                              INTEGRALS FOR ENGINEERS AND PHYSICISTS, 00023240
      C                              FORMULA  900.00 FOR KACC2.GE.0.5        00023250
      C                              FORMULA  900.06 FOR KACC2.LT.0.5        00023260
      C-----------------------------------------------------------------00023270
            DOUBLE PRECISION KACC2,KA,M1,KACC                              00023280
            KA = 1.0D0-KACC2                                               00023290
            IF(KA.GT.0.5D0) GO TO 10                                       00023300
            FLLK=1.0D0+KA*(0.25D0+KA*(0.140625D0+KA*(0.09765625D0+KA*(0.07476800023310
           10664D0+KA*(0.0605621338D0+KA*(0.0508890150D0+KA*(0.0438787937D0+KA00023320
           2*(0.0385653465D0+KA*(0.0343993364D0+KA*(0.0310454012D0+KA*(0.028280023330
           372353D0+KA*(0.0259790743D0+KA*(0.0240191152D0+KA*(0.0223341012D0+K00023340
           4A*0.0208699768D0))))))))))))))                                  00023350
            GO TO 30                                                       00023360
         10 KACC=DSQRT(KACC2)                                              00023370
            IF(KACC.LT.1.0D-04) GO TO 20                                   00023380
            M1=-DLOG(KACC)                                                 00023390
            FLLK=M1*(1.0D0+KACC2*(0.25D0+KACC2*(0.140625D0+KACC2*(0.09765625D000023400
           1+KACC2*(0.0747680664D0+KACC2*(0.0605621338D0+KACC2*(0.050889015  D00023410
           20+KACC2*(0.0438787937 D0+KACC2*(0.0385653465 D0+KACC2*(0.034399336 00023420
           34 D0+KACC2*(0.0310454012 D0+KACC2*(0.0282872353 D0+KACC2*(0.025979 00023430
           40743 D0+KACC2*(0.0240191152 D0+KACC2*0.0223341012 D0)))))))))))))))00023440
           5+1.38629436D0+KACC2*(0.0965735903D0+KACC2*(0.0308851445D0+KACC2*(000023450
           6.0149376004D0+KACC2*(0.0087663122D0+KACC2*(0.0057548877D0+KACC2*(000023460
           7.0040646585D0+KACC2*(0.0030225465D0+KACC2*(0.0023351572D0+KACC2*(000023470
           8.0018580703D0+KACC2*(0.0015135116D0+KACC2*(0.0012565911D0+KACC2*(000023480
           9.0010599297D0+KACC2*(0.0009060596D0+KACC2*0.0007834118D0)))))))))00023490
           T)))                                                            00023500
            FLLK = 2.0D0*FLLK/3.1415926535D0                               00023510
            GO TO 30                                                       00023520
         20 FLLK = 0.0D0                                                   00023530
         30 RETURN                                                         00023540
            END                                                            00023550
            DOUBLE PRECISION FUNCTION FLLE(KACC2)                          00023560
      C-----------------------------------------------------------------00023570
```

```
C                                THIS FUNCTIONSUBROUTINE EVALUATES THE
C                                COMPLETE ELLEPTIC INTEGRAL OF THE SECOND
C                                KIND FROM A SERIES-EXPANSION ACCORDING TO
C                                BYRD AND FRIEDMAN,HANDBOOK OF ELLIPTIC
C                                INTEGRALS FOR ENGINEERS AND PHYSICISTS,
C                                FORMULA  900.07 FOR KACC2.GE.0.5
C                                FORMULA  900.10 FOR KACC2.LT.0.5
C-----------------------------------------------------------------------
      DOUBLE PRECISION KACC2,KACC,KA
      KA = 1.0D0-KACC2
      IF(KA.GT.0.5D0) GO TO 10
      FLLE=1.0D0-KA*(0.25D0+KA*(0.046875D0+KA*(0.01953125D0+KA*(0.010681
     11523D0+KA*(0.0067291260D0+KA*(0.0046262741D0+KA*(0.0033752918D0+KA
     2*(0.0025710231D0+KA*(0.0020234904D0+KA*(0.0016339685D0+KA*0.001347
     30112D0))))))))))
      GO TO 30
   10 KACC=DSQRT(KACC2)
      IF(KACC.LT.1.0D-04)       GO TO 20
      FLLE=1.0D0-0.50D0*KACC2*DLOG(KACC)*(1.0D0+KACC2*(0.375D0+KACC2*(0 
     1.234375D0+KACC2*(0.1708984375D0+KACC2*(0.1345825196D0+KACC2*(0.111
     20305786D0+KACC2*(0.0945081711D0+KACC2*(0.0822727382D0+KACC2*(0.072
     38456536D0+KACC2*(0.0653587393D0+KACC2*(0.0592684931D0+KACC2*(0.054
     42172011D0+KACC2*(0.0499597606D0+KACC2*(0.0463225802D0+KACC2*0.0431
     5792623D0))))))))))))))+0.25D0*KACC2*(1.772588722D0+KACC2*(0.227220
     67707D0+KACC2*(0.0873254817D0+KACC2*(0.0461780856D0+KACC2*(0.028568
     70012D0+KACC2*(0.0194189733D0+KACC2*(0.0140587518D0+KACC2*(0.010648
     89434D0+KACC2*(0.0083455895D0+KACC2*(0.0067166737D0+KACC2*(0.005522
     92888D0+KACC2*(0.0046204936D0+KACC2*(0.0039229304D0+KACC2*(0.003372
     T2549D0+KACC2*0.0029299298D0)))))))))))))))
      FLLE = 2.0D0*FLLE/3.1415926535D0
      GO TO 30
   20 FLLE = 2.0D0/3.1415926535D0
   30 RETURN
      END
      FUNCTION FLMBDA(DR,C,ELLK,ELLE,KACC2)
C-----------------------------------------------------------------------
C                                THIS FUNCTIONSUBROUTINE EVALUATES THE
C                                HEUMAN'S-LAMBD' "UNCTION FROM A SERIES-
C                                EXPANSION ACCORDING TO
C                                BYRD AND FRIEDMAN,HANDBOOK OF ELLIPTIC
C                                INTEGRALS FOR ENGINEERS AND PHYSICISTS,
C                                FORMULA  904.00
C                                USE IS MADE OF THE COMPLETE ELLIPTIC INTE-
C                                GRALS OF THE FIRST AND SECOND KIND ELLK
C                                AND ELLE EVALUATED BY FLLK AND FLLE.
C-----------------------------------------------------------------------
      DOUBLE PRECISION DR,DASIN,SUM,PHI,DS,DC,A,T,AI,KACC2,TWAI,DAR,ELLK
     1,ELLE,E,K,C
      DAR = DABS(1.0D0-DR)
      IF(C.LT.DAR) GO TO 10
      DASIN = DAR/C
      PHI = 1.5707963268D0-DATAN(DASIN)
      GO TO 30
```

```
   10 DASIN = C/DAR
      IF(C.LT.(0.1D-05*DAR)) GO TO 20
      PHI = DATAN(DASIN)
      GO TO 30
   20 PHI = DASIN
   30 IF(DABS(PHI-1.5707963268D0).GT.1.0D-6) GO TO 40
      FLMBDA=1.0
      GO TO 60
   40 DS=DSIN(PHI)
      DC=DCOS(PHI)
      E = ELLE
      K = ELLK
      FLMBDA=PHI*E
      T=0.5D0*(PHI-DS*DC)
      A=0.5D0*KACC2
      SUM=A*T*(2.0D0*K-E)
      IF(SUM.LT.1.0D-07) GO TO 60
      I=1
   50 FLMBDA = FLMBDA-SNGL(SUM)
      I=I+1
      AI=I
      TWAI=2.0D0*AI-1.0D0
      T=0.5D0*T*TWAI/AI-0.5D0*DC*(DS**TWAI)/AI
      A=0.5D0*A*(TWAI-2.0D0)*KACC2/AI
      SUM=A*T*(2.0D0*AI*K-TWAI*E)
      IF(SUM.GT.1.0D-07) GO TO 50
   60 RETURN
      END
      SUBROUTINE ASS(ACC,R)
C-----------------------------------------------------------------------
C                                THIS SUBROUTINE COMPUTES THE LIPSCHITZ-
C                                HANKEL INTEGRALS I(I,J,K) FROM EXPRESSI-
C                                ONS IN EARLIER EVALUATED ELLIPTIC FUNCTI-
C                                ONS OF THE FIRST AND SECOND KIND,ELLK AND
C                                ELLE,AND HEUMAN'S-LAMBDA FUNCTION,ALMBDA.
C                                REFERENCE
C                                EASON,NOBLE AND SNEDDON,CERTAIN INTEGRALS
C                                OF LIPSCHITZ-HANKEL TYPE INVOLVING PRO-
C                                DUCTS OF BESSEL FUNCTIONS,PHILOSOPHICAL
C                                TRANSACTIONS,VOL 247,SERIES A935,APRIL
C                                1955,PP 529-546.
C                                           F10M1=I(1,0;-1)
C                                           F100 =I(1,0;0)
C                                           F101 =I(1,0;1)
C                                           F11M2=I(1,1;-2)
C                                           F11M1=I(1,1;-1)
C                                           F110 =I(1,1;0)
C                                           F111 =I(1,1;1)
C-----------------------------------------------------------------------
      COMMON/CONST/C,ELLE,ELLK,ALMBDA
      COMMON/CNTING/F10M1,F100,F101,F11M2,F11M1,F110,F111
      DOUBLE PRECISION DR,C,DEPR,DEMR,DC2,DRT2,DRT,DAD,DR2,DRC2,DRRT,
     1DEMRR,ELLE,ELLK
```

```
      EC = SNGL(C)
      IF(R.LT.ACC) GO TO 20
      EMR = 1.0-R
      EPR = 1.0+R
      C2 = EC*EC
      RT2 = C2+EPR*EPR
      RT = SQRT(RT2)
      R2 = R*R
      EMRR = 1.0-R2
      DR = DBLE(R)
      DEPR = 1.0D0+DR
      DEMR = 1.0D0-DR
      DC2 = C*C
      DRT2 = DC2+DEPR*DEPR
      DRT = DSQRT(DRT2)
      DAD = DC2+DEMR*DEMR
      DR2 = DR*DR
      DRC2 = DR2+DC2
      DRRT = DR*DRT
      DEMRR = 1.0D0-DR2
      F101 = 0.5D0*(ELLE*(1.0D0-DRC2)/(DAD*DRT)+ELLK/DRT)
      F110 = DRT*(ELLK*(1.0D0+DRC2)/DRT2-ELLE)/(2.0D0*DR)
      F111 = C*(ELLE*(1.0D0+DRC2)/DAD-ELLK)/(2.0D0*DRRT)
      F10M1 = 0.5D0*ELLE*DRT
      F100 = -0.5D0*C*ELLK/DRT
      F11M2 = -C*(C*ELLE*DRT/(4.0D0*DR)-C*ELLK*(1.0D0+DR2+0.5D0*DC2)/
     1(2.0D0*DRRT))/3.0D0+DR*(DRT*ELLE/2.0D0+DEMRR*ELLK/(2.0D0*DRT))/
     23.0D0+(ELLE*DRT/(2.0D0*DR)-DEMRR*ELLK/(2.0D0*DRRT))/3.0D0
      F11M1 = 0.5D0*(0.5D0*ELLE*C*DRT/DR-ELLK*C*(1.0D0+DR2+0.5D0*DC2)/
     1DRRT)
      HLP = R
      IF(R.GT.1.0) HLP=1.0/R
      IF(ABS(EMR).LT.ACC) GO TO 10
      F10M1 = F10M1+0.5*(SNGL(ELLK)*EMRR/RT+SIGN(EC*ALMBDA,EMR))
      F100 = F100+0.5*SIGN(ALMBDA,-EMR)
      F11M2 = F11M2-EC*ALMBDA*SIGN(1.0,EMR)*EMRR/(4.0*R)
      F11M2 = F11M2-EC*(HLP/2.0+HLP)/3.0
      F11M1 = F11M1+SIGN(0.25,EMR)*EMRR*ALMBDA/R+0.5*HLP
      IF(R.GT.1.0) GO TO 30
      F10M1 = F10M1-EC
      F100 = F100+1.0
      GO TO 30
   10 F10M1 = F10M1-0.5*EC
      F100 = F100+0.5
      F11M2 = F11M2-0.5*EC
      F11M1 = F11M1+0.5*HLP
      GO TO 30
   20 AD = 1.0+EC*EC
      RT = SQRT(AD)
      ADRT = AD*RT
      F101 = 1.0/ADRT
      F110 = 0.5/ADRT
      F111 = 1.5*EC/(AD*ADRT)
```

```
      F10M1 = RT-EC
      F100 = 1.0-EC/RT
      F11M2 = 0.5*(RT-EC)
      F11M1 = 0.5*(1.0-EC/RT)
   30 RETURN
      END
      SUBROUTINE INGRAL(IL,INTV,INTT,INT)
C-----------------------------------------------------------------------
C                                 THIS SUBROUTINE CONTROLS THE SIMULTANEOUS
C                                 COMPUTATION OF A GROUP OUT OF THE 17 INTE-
C                                 GRALS.
C                                 IL=1,THE GROUP WITH J0(XR)J1(X) IN INTEGR.
C                                 IL=2,THE GROUP WITH J1(XR)J1(X) IN INTEGR.
C                                 THE ELEMENTS OF INTV ARE THE NUMBERS OF
C                                 REQUIRED INTEGRALS OF THE GROUP,
C                                 INTT IS THE TOTAL NUMBER OF REQUIRED INTE-
C                                 GRALS IN THE GROUP.
C                                 THE SET OF COMPUTED INTEGRALS IS DELIVE-
C                                 RED IN INT.
C                                 ACTUAL INTEGRATION BY MEANS OF A GAUSS-
C                                 QUADRATURE IS PERFORMED BY CALLING QUAD.
C                                 INTEGRATION PROCEEDS BY QUADRATURE OVER
C                                 INTERVALS FROM ONE BESSEL ZERO TO THE
C                                 NEXT. FROM THE ORIGIN TO THE FIRST BESSEL-
C                                 ZERO A LEGENDRE-GAUSS QUADRATURE OF ORDER
C                                 8,OBTAINING DESIRED ACCURRACY BY SUBSE-
C                                 QUENT SUBDIVISION OF THE INTERVAL.
C                                 FROM THE FIRST BESSELZERO ON A JACOBI-
C                                 GAUSS QUADRATURE,OBTAINING DESIRED ACCU-
C                                 RACY BY SUBSEQUENT RAISING THE ORDER
C                                 STARTING WITH THE 4TH ORDER.
C                                 INTEGRATION STOPS AS SOON AS TWO SUBSE-
C                                 QUENT INTERVALS  DO NOT CONTRIBUTE
C                                 SIGNIFICANTLY.
C                                 INTEGRATION STOPS PREMATURELY IF#
C                                 -IN THE FIRST INTERVAL MORE THAN 30 SUB-
C                                  DIVISIONS ARE NEEDED.
C                                 -IN THE FOLLOWING INTERVALS EVEN THE 15TH
C                                  ORDER IS NOT ACCURATE ENOUGH.
C                                 -EVEN THE 149-TH(298 TH)INTERVAL DOES
C                                  GIVE A NON-NEGIGIBLE CONTRIBUTION.
C-----------------------------------------------------------------------
      INTEGER ALFA,ORDER,INTV(10),INTV2(10),INTV3(10),KK(10),BETA
      REAL MIDPNT,LOWER,LOAD,NU,ACCUR(3),K5,COMP(10),FIRST(10),
     1SECOND(10),INT(17),RES(10)
      COMMON/ASDT/LAYER,NLAYS,M,R,Z,NU(10),ACCUR,LOAD,HOSTRS,NZEROS,H(9)
     1,K5(10),E(10),AL(9),THICK(9),RADIUS(10)
      COMMON/GEDATA/BZEROS(149,2),ZEROS(298)
      COMMON/CNTING/F10M1,F100,F101,F11M2,F11M1,F110,F111
      COMMON/TAPE/NOUT
      NTELL = 0
      NINT = 7
      IF(IL.EQ.2) NINT = 10
```

```
      DO 1000 I = 1,NINT
         KK(I) = 0
 1000 CONTINUE
      IF(LAYER.NE.1) GO TO 2000
C-----------------------------------------------------------------------
C                                      CALCULATION OF THE ASYMPTOTIC PART OF THE
C                                      INTEGRALS,FOR POINTS IN THE TOPLAYER ONLY.
C-----------------------------------------------------------------------
      DO 1190 I = 1,NINT
         K = INTV(I)
         IF(K.EQ.0) GO TO 1190
         GO TO (1010,1020,1030,1040,1050,1060,1070,1080,1090,1100,1110,
     1   1120,1130,1140,1150,1160,1170),K
 1010    INT(K) = F100+Z*F101
         GO TO 1190
 1020    INT(K) = F100
         GO TO 1190
 1030    INT(K) = -2.0*(1.0-NU(1))*F10M1-Z*F100
         GO TO 1190
 1040    INT(K) = (1.0-2.0*NU(1))*F11M1-Z*F110
         GO TO 1180
 1050    INT(K) = Z*F111
         GO TO 1190
 1060    INT(K) = F100-Z*F101
         GO TO 1190
 1070    INT(K) = -2.0*(1.0-NU(1))*F10M1+Z*F100
         GO TO 1190
 1080    INT(K) = -2.0*(1.0-NU(1))*F110+Z*F111
         GO TO 1190
 1090    INT(K) = -F110
         GO TO 1190
 1100    INT(K) = F11M1-Z*F110
         GO TO 1180
 1110    INT(K) = F11M1
         GO TO 1180
 1120    INT(K) = -2.0*(1.0-NU(1))*F11M2+Z*F11M1
         GO TO 1180
 1130    INT(K) = -F100
         GO TO 1190
 1140    INT(K) = F10M1
         GO TO 1190
 1150    INT(K) = F110
         GO TO 1190
 1160    INT(K) = -F11M1
         GO TO 1180
 1170    INT(K) = F11M2
 1180    IF(R.GE.ACCUR(1)) INT(K) = INT(K)/R
 1190 CONTINUE
      IF(NLAYS.EQ.1) GO TO 3140
C-----------------------------------------------------------------------
C                                      INTEGRATION FROM THE ORIGIN TO THE FIRST
C                                      BESSELZERO.
C-----------------------------------------------------------------------
```

```
 2000 INTT2 = INTT
      INTT3 = INTT
      DO 2010 J = 1,NINT
         INTV2(J) = INTV(J)
         INTV3(J) = INTV(J)
 2010 CONTINUE
      UPPER = ZEROS(1)
      ALFA = 0
      BETA = 0
      IRES = 0
      DELTA = 0.5*ZEROS(1)
 2020 LOWER = UPPER-DELTA
      IF(LOWER-ACCUR(1)) 2030,2030,2040
 2030 ALFA = -1
      LOWER = 0.0
 2040 IF(IRES.EQ.1) GO TO 2050
      CALL QUAD(IL,INTV3,LOWER,UPPER,16,COMP,NTELL)
      IF(NTELL.NE.0) GO TO 3180
      GO TO 2070
 2050 DO 2060 J = 1,NINT
         COMP(J) = RES(J)
 2060 CONTINUE
      IRES = 0
 2070 MIDPNT = 0.5*(LOWER+UPPER)
      CALL QUAD(IL,INTV3,LOWER,MIDPNT,16,FIRST,NTELL)
      IF(NTELL.NE.0) GO TO 3180
      CALL QUAD(IL,INTV3,MIDPNT,UPPER,16,SECOND,NTELL)
      IF(NTELL.NE.0) GO TO 3180
      DO 2090 J = 1,NINT
         IF(INTV3(J).EQ.0) GO TO 2090
         IF(ABS(COMP(J)).LT.ACCUR(2)) GO TO 2080
         IF(ABS((COMP(J)-FIRST(J)-SECOND(J))/COMP(J)).LT.ACCUR(3))
     1   GO TO 2080
         GO TO 2090
 2080    INTT3 = INTT3-1
         IF(LOWER.GT.ACCUR(1)) GO TO 2090
         INTT2 = INTT2-1
         INTV3(J) = 0
 2090 CONTINUE
      IF(INTT3.EQ.0) GO TO 2110
      ALFA = 0
      LOWER = MIDPNT
      DELTA = 0.5*DELTA
      BETA = BETA+1
      IF(BETA.GT.30) GO TO 2150
C-----------------------------------------------------------------------
C                    ARRIVAL HERE MEANS THAT THE INTEGRAND IS TOO
C                    IRREGULAR TO GET INTEGRATED OVER THE REGION FROM
C                    THE ORIGIN TO THE FIRST BESSEL ZERO.
C-----------------------------------------------------------------------
      IRES = 1
      DO 2100 J = 1,NINT
         COMP(J) = SECOND(J)
```

```
            RES(J) = FIRST(J)
 2100 CONTINUE
      INTT3 = INTT2
      GO TO 2070
 2110 DO 2120 J = 1,NINT
         K = INTV2(J)
         IF(K.EQ.0) GO TO 2120
         INT(K) = INT(K)+FIRST(J)+SECOND(J)
         IF(INTV3(J).NE.0) GO TO 2120
         INTV2(J) = 0
 2120 CONTINUE
      UPPER = LOWER
      INTT3 = INTT2
      IF(ALFA) 3000,2140,2130
 2130 DELTA = DELTA*2.0
      BETA = BETA-1
 2140 ALFA = ALFA+1
      GO TO 2020
 2150 WRITE(NOUT,9040)
      GO TO 3180
C----------------------------------------------------------------------
C                             INTEGRATION OVER THE REMAINING INTERVALS.
C----------------------------------------------------------------------
 3000 IFIN = NZEROS-1
      DO 3010 J = 1,NINT
         INTV2(J) = INTV(J)
 3010 CONTINUE
      INTT2 = INTT
      DO 3130 IBESS = 1,IFIN
         DO 3020 J = 1,NINT
            INTV3(J) = INTV2(J)
            FIRST(J) = 0.0
 3020    CONTINUE
         INTT3 = INTT2
         DO 3070 ORDER = 4,15
            CALL QUAD(IL,INTV3,ZEROS(IBESS),ZEROS(IBESS+1),ORDER,SECOND,
     1      NTELL)
            IF(NTELL.NE.0) GO TO 3180
            DO 3060 J = 1,NINT
               K = INTV3(J)
               IF(K.EQ.0) GO TO 3060
               IF(ABS(INT(K)).LT.0.01) GO TO 3030
               IF(ABS((FIRST(J)-SECOND(J))/INT(K)).LT.0.1*ACCUR(3))
     1         GO TO 3040
               GO TO 3050
 3030          IF(ABS(FIRST(J)-SECOND(J)).GE.0.1*ACCUR(2)) GO TO 3050
 3040          INTV3(J) = 0
               INTT3 = INTT3-1
               GO TO 3060
 3050          FIRST(J) = SECOND(J)
 3060       CONTINUE
            IF(INTT3.EQ.0) GO TO 3080
 3070    CONTINUE
```

```
          WRITE(NOUT,9020)
          WRITE(NOUT,9050) ZEROS(IBESS)
C-----------------------------------------------------------------------
C                   ARRIVAL HERE MEANS THAT THE DESIRED ACCURACY CANNOT
C                   BE MET BY MEANS OF THE AVAILABLE GAUSS-JACOBI
C                   POLYNOMINALS.
C-----------------------------------------------------------------------
          GO TO 3180
  3080    DO 3120 J = 1,NINT
             K = INTV2(J)
             IF(K.EQ.0) GO TO 3120
             INT(K) = INT(K)+SECOND(J)
             IF(ABS(INT(K)).LT.0.01) GO TO 3090
             IF(ABS(SECOND(J)/INT(K)).LT.0.1*ACCUR(3)) GO TO 3110
             GO TO 3100
  3090       IF(ABS(SECOND(J)).LT.0.1*ACCUR(2)) GO TO 3110
  3100       KK(J) = 0
             GO TO 3120
  3110       KK(J) = KK(J)+1
             IF(KK(J).LT.2) GO TO 3120
             INTV2(J) = 0
             INTT2 = INTT2-1
  3120    CONTINUE
          IF(INTT2.EQ.0) GO TO 3140
  3130 CONTINUE
       WRITE(NOUT,9030)
       WRITE(NOUT,9050) ZEROS(IFIN)
C-----------------------------------------------------------------------
C                   ARRIVAL HERE MEANS THAT ALL AVAILABLE BESSEL ZEROS
C                   HAVE BEEN EXHAUSTED BECAUSE OF ILL CONVERGENCE OF
C                   THE INTEGRALS.
C-----------------------------------------------------------------------
       GO TO 3180
  3140 DO 3170 J = 1,NINT
          K = INTV(J)
          IF(K.EQ.0) GO TO 3170
          IF(K-5) 3150,3150,3160
  3150    INT(K) = INT(K)*LOAD
          GO TO 3170
  3160    INT(K) = INT(K)*HOSTRS
  3170 CONTINUE
       RETURN
  3180 WRITE(NOUT,9010) (INTV3(J),J=1,NINT)
       GO TO 3140
  9010 FORMAT(' DURING CALCULATION OF INTEGRALS',10I3)
  9020 FORMAT(' SUSPEND PROGRAM GAUSS POLYS EXHAUSTED')
  9030 FORMAT(' SUSPEND PROGRAM BESSEL ZEROS EXHAUSTED')
  9040 FORMAT(' SUSPEND PROGRAM STEPSIZE FIRST INTERVAL TOO SMALL')
  9050 FORMAT(' AT THE VALUE #',E11.4,' FOR THE INTEGRATION VARIABLE')
       END
       SUBROUTINE QUAD (IL,INTV,ALO,UP,NGAUSS,FSC,NTELL)
C-----------------------------------------------------------------------
C                               THIS SUBROUTINE CALCULATES FOR THE SET
```

```
C                              INTV THE INTEGRALS OF THE CORRESPONDING
C                              FUNCTIONS IGRAND BETWEEN THE LIMITS ALO
C                              AND UP BY USING A GAUSS QUADRATURE OF
C                              ORDER NGAUSS.
C                              -FOR NGAUSS=16 A LEGENDRE-GAUSS QUADRATU-
C                               RE OF ORDER 8.
C                              -FOR NGAUSS.LT.16 A JACOBI-GAUSSQUADRATU-
C                               RE.
C                              THE ABSCISSAE AND WEIGHTS OF BOTH ARE
C                              STORED AS AGAUSS AND HGAUSS IN THE BLOCK
C                              DATA.
C                              THE SET OF INTEGRANDS IS COMPUTED DURING
C                              SUBSEQUENT CALLING IN OF*
C                                        SUBROUTINES  MATRIX
C                                                     FPIGRA
C                                       AND FUNCTION  IGRAND
C                              THE SET OF RESULTING INTEGRALS IS
C                              DELIVERED IN FSC
C-----------------------------------------------------------------------
      INTEGER INTV(10)
      REAL IGRAND,FSC(10)
      COMMON/GAUSS/AGAUSS(16,16),HGAUSS(16,16)
      NINT = 7
      IF(IL.EQ.2) NINT = 10
      DO 10 J = 1,NINT
         K = INTV(J)
         IF(K.EQ.0) GO TO 10
         FSC(J) = 0.0
   10 CONTINUE
      LABEL = 0
      IF(IL.EQ.2) GO TO 20
      IF((INTV(1)+INTV(2)+INTV(3)).GT.0) LABEL = LABEL+1
      IF((INTV(4)+INTV(5)).GT.0) LABEL = LABEL+2
      IF((INTV(6)+INTV(7)).GT.0) LABEL = LABEL+4
      GO TO 30
   20 IF((INTV(1)+INTV(2)).GT.0) LABEL = LABEL+1
      IF((INTV(3)+INTV(4)+INTV(5)+INTV(6)+INTV(7)).GT.0) LABEL = LABEL+2
      IF((INTV(8)+INTV(9)+INTV(10)).GT.0) LABEL = LABEL+4
   30 F1 = 0.5*(UP-ALO)
      F2 = 0.5*(UP+ALO)
      IGAUSS = NGAUSS
      IF(NGAUSS.EQ.16) IGAUSS=8
      DO 50 I = 1,IGAUSS
         X = F1*AGAUSS(I,NGAUSS)+F2
         CALL MATRIX (X,LABEL,NTELL)
         IF(NTELL.EQ.1) RETURN
         CALL FPIGRA (IL,X)
         DO 40 J = 1,NINT
            K = INTV(J)
            IF(K.EQ.0) GO TO 40
            FSC(J) = FSC(J)+HGAUSS(I,NGAUSS)*IGRAND(X,K)
   40    CONTINUE
   50 CONTINUE
```

```
      DO 60 J = 1,NINT
         K = INTV(J)
         IF(K.EQ.0) GO TO 60
         FSC(J) = FSC(J)*F1
   60 CONTINUE
   70 RETURN
      END
      SUBROUTINE MATRIX (X,LABL,NTELL)
C-----------------------------------------------------------------------
C                                  THIS SUBROUTINE COMPUTES THE SET OF CHA-
C                                  RACTERISTIC-FUNCTIONS T0,V0,S0,U0,T1,V1,
C                                  S1,U1,TQ1 AND SQ1 FOR THE VALUE X OF THE
C                                  INTEGRATION-PARAMETER. USE IS MADE OF
C                                  CONSTANTS CALCULATED IN MACON1 AND MA2CON.
C                                  THEY WERE STORED IN COMMON/INDATA/.
C                                  CHARACTERISTIC-FUNCTION VALUES ARE DELIVE-
C                                  RED IN COMMON/IGRAN/.
C                                  LABL DETERMINES WHICH CHARACTERISTIC-
C                                  FUNCTIONS ARE NEEDED#
C                                  -LABL=1#T0,V0,S0,U0
C                                  -LABL=2#T1,V1,S1,U1
C                                  -LABL=3#T0,V0,S0,U0,T1,V1,S1,U1
C                                  -LABL=4#TQ1,SQ1
C                                  -LABL=5#T0,V0,S0,U0,TQ1,SQ1
C                                  -LABL=6#T1,V1,S1,U1,TQ1,SQ1
C                                  -LABL=7#T0,V0,S0,U0,T1,V1,S1,U1,TQ1,SQ1
C                                  SUBROUTINE IS INTERRUPTED AND RETURNED
C                                  WITH NTELL=1 WHEN SOLUTION BECOMES TOO
C                                  INACCURATE BECAUSE OF ILL MATRIX-CONDI-
C                                  ON DURING INVERSION.
C-----------------------------------------------------------------------
      REAL LOAD,NU,W(4,4,9),P(4,2),PP(2,2),K1,K2,K5,K6,II,NJ(2,2,9),KK6,
     1ACCUR(3),NP(2,10),NJ2(9),P3(2),NP2(10),K4(10)
      COMMON/ASDT/LAYER,NLAYS,M,R,Z,NU(10),ACCUR,LOAD,HOSTRS,NZEROS,H(9)
     1,K5(10),E(10),AL(9),THICK(9),RADIUS(10)
      COMMON/INDATA/XMAX,  A1(9),B1(9),C1(9),D(9),EE(9),F(9),G(9),H1(9),
     1II(9),K1(9),K2(9),K6(10),BE(9),BU(9),BUU(9),BMU(9),B2U(9),B2UU(9),
     2J2(9),J1,T2(10),SS(2,10),G012(9),G021(9),G022(9),G122(9),
     3H012(9),H022(9),H122(9),D012(9),D022(9),C011(9),C012(9),E012(9),
     4F012(9),F112(9),F022(9),CC(4,2,9),DD(2,2,9),FF(2,2,9),GG(2,2,10),
     5HH(2,2,10),RR(4,2,10),DD2(9),G20(9),G21(9),H20(9),H021(9),GG2(10),
     6HH2(10),Q011(9),Q111(9),Q012(9),Q112(9),Q212(9),Q022(9),Q122(9),
     7QF0(9),QF1(9),Z011,Z111,Z211,Z012,Z112,Z212,Z312,Z021,Z121,Z022,
     8Z122,Z222,K4
      COMMON/IGRAN/T0,V0,S0,U0,T1,V1,S1,U1,TQ1,SQ1,FPIGR,EX1,EX2
      COMMON/TAPE/NOUT
      LABEL = LABL
      IF(LABEL.LT.4) GO TO 1000
      IF(X.LT.XMAX) GO TO 100
C-----------------------------------------------------------------------
C                                  ASYMPTOTIC EVALUATION OF TQ1 AND SQ1
C                                  FOR X.GE.XMAX.
C-----------------------------------------------------------------------
```

```
      TQ1 = 1.0
      IF(LAYER.EQ.1) GO TO 30
      J = LAYER-1
      DO 20 K = 1,J
         TQ1=TQ1*2.0*(1.0-AL(K))/((1.0-AL(K))*(1.0+K1(K))+0.5*AL(K)*X)
   20 CONTINUE
   30 SQ1=TQ1*(0.5*AL(LAYER)*X-(1.0-AL(LAYER))*K2(LAYER))/((1.0-AL(LAYER
     1   )) *(1.0+K1(LAYER))+0.5*AL(LAYER)*X)
      LABEL = LABEL-4
      GO TO 1000
C-----------------------------------------------------------------------
C                          CALCULATION OF TQ1 AND SQ1 FOR X.LT.XMAX
C-----------------------------------------------------------------------
  100 IF(J1.EQ.0) GO TO 120
      DO 110 J = 1,J1
         GG2(J+1) = G20(J)-G21(J)*X
         HH2(J+1) = H20(J)+G21(J)*X
  110 CONTINUE
  120 DO 150 K = 1,M
         IF(J1.EQ.0) GO TO 140
         DO 130 I = 1,J1
            IF(J2(I).EQ.K) GO TO 150
  130    CONTINUE
  140    W1 = 0.5*(1.0+K1(K))
         W2 =-0.5*K2(K)
         W3 = BE(K)*0.25*X
         NJ(1,1,K) = W1+W3
         NJ(1,2,K) = W2-W3
         NJ(2,1,K) = W2+W3
         NJ(2,2,K) = W1-W3
  150 CONTINUE
      J5 = J1+1
      DO 300 MM = 1,J5
         N = J5+1-MM
         IF(N-1) 160,160,170
  160    J3 = 1
         GO TO 180
  170    J3 = J2(N-1)+1
  180    IF(J5-N) 190,190,200
  190    J4 = M
         GO TO 210
  200    J4 = J2(N)-1
  210    IF(J3.GT.J4) GO TO 240
         DO 230 IJ = J3,J4
         IK = J4+J3-IJ
         IL = IK+1
      EXPO=-X*T2(IL)
      IF(EXPO.LT.-70.0)GO TO 212
      EXP1=EXP(EXPO)*SS(1,IL)
      GO TO 214
  212 EXP1=0.0
  214 DO 220 I=1,2
  220       SS(I,IK) = NJ(I,1,IK)*EXP1+NJ(I,2,IK)*SS(2,IL)
```

```
  230    CONTINUE
  240    NN = N-1
      EXPO=-X*T2(J3)
      IF(EXPO.LT.-70.0)GO TO 242
      EXP2=EXP(EXPO)
      GO TO 244
 242  EXP2=0.0
 244  PROD=GG2(N)*SS(1,J3)*EXP2
         P2 = PROD+HH2(N)*SS(2,J3)
C-----------------------------------------------------------------------
C                                TEST MATRIX-CONDITION BEFORE INVERSION.
C-----------------------------------------------------------------------
         IF(ABS(P2).LT.1.0E-7*ABS(PROD)/ACCUR(3)) GO TO 2000
         PP2 = 1.0/P2
         IF(N.EQ.1) GO TO 310
         NJ2(NN) = PP2*DD2(NN)
         DO 270 I = 1,2
  270       P3(I) = SS(I,J3)*NJ2(NN)
      PP2=P3(1)*EXP2-P3(2)
         SS(1,J3-1) = PP2+1.0
         SS(2,J3-1) = 1.0
  300 CONTINUE
  310 IF(NTELL.EQ.2) RETURN
      NP2(1) = -PP2
      IF(J1.EQ.0) GO TO 390
      DO 350 I = 1,J1
         J = J1+1-I
         IF(LAYER.GT.J2(J)) GO TO 360
  350 CONTINUE
      J5 = 1
      GO TO 390
  360 DO 380 I = 1,J
         NP2(I+1) = NJ2(I)*NP2(I)
  380 CONTINUE
      J5 = J+1
  390 J = LAYER
      SQ1 = SS(1,J)*NP2(J5)
      TQ1 = SS(2,J)*NP2(J5)
      LABEL = LABEL-4
C-----------------------------------------------------------------------
C                            ASYMPTOTIC EVALUATION OF T0,V0,S0,U0,T1,
C                            V1,S1 AND U1 FOR X.GE.XMAX.
C-----------------------------------------------------------------------
 1000 IF(LABEL.EQ.0) RETURN
      IF(X.LT.XMAX) GO TO 1100
      L = LAYER
      X2 = X*X
      X3 = X2*X
      IF(L.EQ.NLAYS) GO TO 1010
      Z11 = Z011+X*Z111+X2*Z211
      Z12 = Z012+X*Z112+X2*Z212+X3*Z312
      Z21 = Z021+X*Z121
      Z22 = Z022+X*Z122+X2*Z222
```

```
 1010 IF(LABEL.GT.1) GO TO 1030
 1020 IF(LABEL.EQ.0) RETURN
      NP(1,1) = 2.0*NU(1)
      NP(2,1) = 1.0
      GO TO 1040
 1030 NP(1,1) = 1.0-2.0*NU(1)
      NP(2,1) = -1.0
 1040 PQF = 1.0
      IF(L.EQ.1) GO TO 1060
      DO 1050 K = 2,L
         J = K-1
         PQF = PQF*K6(J)/(QF0(J)+QF1(J)*X)
         W1 = -AL(J)*X
         W9 = H(J)*X
         NP(1,K) = NP(1,J)*(Q011(J)+Q111(J)*X+W1*W9)+NP(2,J)*(Q012(J)
     1   +Q112(J)*X+Q212(J)*X2+W1*W9*W9)
 1050    NP(2,K) = -W1*NP(1,J)+NP(2,J)*(Q022(J)+Q122(J)*X-W1*W9)
      IF(L.NE.NLAYS) GO TO 1060
      S = 0.0
      U = 0.0
      GO TO 1070
 1060 S = (NP(1,L)*Z11+NP(2,L)*Z12)*PQF/(QF0(L)+QF1(L)*X)
      U = (NP(1,L)*Z21+NP(2,L)*Z22)*PQF/(QF0(L)+QF1(L)*X)
 1070 T = NP(1,L)*PQF
      V = NP(2,L)*PQF
      IF(LABEL.GT.1) GO TO 1080
      S0 = S
      U0 = U
      T0 = T
      V0 = V
      RETURN
 1080 S1 = S
      U1 = U
      T1 = T
      V1 = V
      LABEL = LABEL-2
      GO TO 1020
C-----------------------------------------------------------------------
C                                      CALCULATION OF T0,V0,S0,U0,T1,V1,S1 AND
C                                      U1 FOR X.LT.XMAX.
C-----------------------------------------------------------------------
 1100 IF(J1.EQ.0) GO TO 1120
      DO 1110 J = 1,J1
         K = J2(J)
         W1 =-AL(K)*X
         W9 = H(K)*X
         CC(1,1,J) = C011(J)+2.0*W9
         CC(1,2,J) = C012(J)+2.0*W9*W9
         CC(2,2,J) = C011(J)-2.0*W9
         DD(1,2,J) = D012(J)+DD(1,1,J)*W9
         DD(2,2,J) = D022(J)+DD(2,1,J)*W9
         FF(1,1,J) = -C011(J)-2.0*W9
         FF(1,2,J) = F012(J)+F112(J)*W9-2.0*W9*W9
```

```
          FF(2,2,J) = F022(J)+2.0*W9
          GG(1,2,J+1) = G012(J)+GG(1,1,J+1)*W9
          GG(2,1,J+1) = G021(J)+W1
          GG(2,2,J+1) = G022(J)+(G021(J)*H(K)+G122(J))*X+W1*W9
          HH(1,2,J+1) = H012(J)+HH(1,1,J+1)*W9
          HH(2,1,J+1) = H021(J)+W1
          HH(2,2,J+1) = H022(J)+H021(J)*W9+H122(J)*X+W1*W9
 1110 CONTINUE
 1120 DO 1150 K=1,M
          IF(J1.EQ.0) GO TO 1140
          DO 1130 I = 1,J1
             IF(J2(I).EQ.K) GO TO 1150
 1130     CONTINUE
 1140     W1 = BMU(K)*X
          W9 = H(K)*X
          W10 = W9*X
          W2 = W10*BE(K)
          W11 = W2*W9
          W3 = W9*C1(K)
          W4 = BE(K)*X
          W5 = BU(K)*X
          W8 = BUU(K)*X
          W7 = C1(K)*W9*W9
          W(1,1,K) = A1(K)+W1-W2
          W(1,2,K) = -EE(K)+F(K)*W9+W8+B2U(K)*W10-W11
          W(1,3,K) = D(K)-W3+W1-W2
          W(1,4,K) = -G(K)+H1(K)*W9-BUU(K)*X-W7+B2UU(K)*W10-W11
          W(2,1,K) = W4
          W(2,2,K) = B1(K)+W5+W2
          W(2,3,K) = C1(K)+W4
          W(2,4,K) = II(K)+W3-W5+W2
          W(3,1,K) = D(K)+W3-W1-W2
          W(3,2,K) = G(K)+H1(K)*W9-W8+W7-B2UU(K)*W10-W11
          W(3,3,K) = A1(K)-W1-W2
          W(3,4,K) = EE(K)+F(K)*W9+W8-B2U(K)*W10-W11
          W(4,1,K) = -C1(K)+W4
          W(4,2,K) = II(K)-W3+W5+W2
          W(4,3,K) = W4
          W(4,4,K) = B1(K)-W5+W2
 1150 CONTINUE
      J5 = J1+1
      PKK6 = 1.0
      DO 1300 MM = 1,J5
          KK6 = 1.0
          N = J5+1-MM
          IF(N-1) 1160,1160,1170
 1160     J3 = 1
          GO TO 1180
 1170     J3 = J2(N-1)+1
 1180     IF(J5-N) 1190,1190,1200
 1190     J4 = M
          GO TO 1210
 1200     J4 = J2(N)-1
```

```
 1210     IF(J3.GT.J4) GO TO 1240
          DO 1230 IJ = J3,J4
             IK = J4+J3-IJ
             IL = IK+1
             KK6 = KK6*K6(IK)
             IF(IK.EQ.LAYER) PKK6 = KK6
          EXPO=-X*T2(IL)
          IF(EXPO.LT.-70.0)GO TO 1212
          EXP1=EXP(EXPO)
          GO TO 1214
 1212     EXP1=0.0
 1214     DO 1220 I=1,4
                DO 1220 K = 1,2
 1220                RR(I,K,IK)=(W(I,1,IK)*RR(1,K,IL)+W(I,2,IK)*RR(2,K,IL))
      1      *EXP1+W(I,3,IK)*RR(3,K,IL)+W(I,4,IK)*RR(4,K,IL)
 1230     CONTINUE
 1240     N = N-1
       EXPO=-X*T2(J3)
       IF(EXPO.LT.-70.0)GO TO 1242
       EXP2=EXP(EXPO)
       GO TO 1244
1242   EXP2=0.0
1244   DO 1250 I=1,2
             DO 1250 K = 1,2
 1250           P(I,K) = (GG(I,1,N)*RR(1,K,J3)+GG(I,2,N)*RR(2,K,J3))
      1      *EXP2+HH(I,1,N)*RR(3,K,J3)+HH(I,2,N)*RR(4,K,J3)
          PROD = P(1,1)*P(2,2)
          DET = PROD-P(1,2)*P(2,1)
C----------------------------------------------------------------------
C                                TEST MATRIX.CONDITION BEFORE INVERSION.
C----------------------------------------------------------------------
          IF(ABS(DET).LT.1.0E-7*ABS(PROD)/ACCUR(3)) GO TO 2000
          QKK6 = KK6/DET
          PP(1,1) = P(2,2)*QKK6
          PP(1,2) = -P(1,2)*QKK6
          PP(2,1) = -P(2,1)*QKK6
          PP(2,2) = P(1,1)*QKK6
           IF(N.EQ.1) GO TO 1310
       DO 1260 I=1,2
             DO 1260 K = 1,2
 1260           NJ(I,K,NN)=PP(I,1)*DD(1,K,NN)+PP(I,2)*DD(2,K,NN)
          DO 1270 I = 1,4
             DO 1270 K = 1,2
 1270           P(I,K) =(RR(I,1,J3)*NJ(1,K,NN)+RR(I,2,J3)*NJ(2,K,NN))/KK6
          DO 1280 I = 1,2
          PP(1,I)=(P(1,I)+E012(NN)*P(2,I))*EXP2+FF(1,1,NN)
      1      *P(3,I)+FF(1,2,NN)*P(4,I)
1280      PP(2,I)=P(2,I)*EXP2+FF(2,1,NN)*P(3,I)+FF(2,2,NN)*P(4,I)
          DO 1290 I = 1,2
             DO 1290 K = 1,2
 1290           RR(I,K,J3-1) = CC(I,K,NN)+PP(I,K)
          RR(3,1,J3-1) = 1.0
          RR(3,2,J3-1) = 0.0
```

```
            RR(4,1,J3-1) = 0.0
            RR(4,2,J3-1) = 1.0
 1300 CONTINUE
 1310 IF(NTELL.EQ.2) GO TO 100
      IF(LABEL.GT.1) GO TO 1330
 1320 IF(LABEL.EQ.0) RETURN
      NP(1,1) = PP(1,1)
      NP(2,1) = PP(2,1)
      GO TO 1340
 1330 NP(1,1) = PP(1,2)
      NP(2,1) = PP(2,2)
 1340 IF(J1.EQ.0) GO TO 1390
      DO 1350 I = 1,J1
         J = J1+1-I
         IF(LAYER.GT.J2(J)) GO TO 1360
 1350 CONTINUE
      J5 = 1
      GO TO 1390
 1360 DO 1380 I = 1,J
         IH = I+1
         DO 1370 K = 1,2
 1370       NP(K,IH) = NJ(K,1,I)*NP(1,I)+NJ(K,2,I)*NP(2,I)
 1380 CONTINUE
      J5 = J+1
 1390 J = LAYER
      S =(RR(1,1,J)*NP(1,J5)+RR(1,2,J)*NP(2,J5))/PKK6
      U =(RR(2,1,J)*NP(1,J5)+RR(2,2,J)*NP(2,J5))/PKK6
      T =(RR(3,1,J)*NP(1,J5)+RR(3,2,J)*NP(2,J5))/PKK6
      V =(RR(4,1,J)*NP(1,J5)+RR(4,2,J)*NP(2,J5))/PKK6
      IF(LABEL.GT.1) GO TO 1400
      T0 = T
      S0 = S
      U0 = U
      V0 = V
      RETURN
 1400 S1 = S
      T1 = T
      U1 = U
      V1 = V
      LABEL = LABEL-2
      GO TO 1320
C-----------------------------------------------------------------------
C                                ARRIVAL HERE MEANS THAT SOLUTION OF THE
C                                CHARACTERISTIC FUNCTIONS HAS BEEN STOPPED
C                                PREMATURELY BECAUSE OF ILL MATRIX CONDI-
C                                TION MET DURING SOLUTION PROCESS.
C-----------------------------------------------------------------------
 2000 WRITE(NOUT,9000)X
      NTELL = 1
      RETURN
 9000 FORMAT(' ILL-CONDITIONED DETERMINANT FOR X=',E15.7)
      END
      SUBROUTINE FPIGRA (IL,X)
```

```
C-----------------------------------------------------------------------
C                                  THIS SUBROUTINE COMPUTES THE BESSELFUNC-
C                                  TION-PART OF THE INTEGRANDS FOR THE
C                                  INTEGRALS COMPUTED IN INGRAL.
C                                  FOR IL=1 THIS PART IS# JO(XR)*J1(X)
C                                  FOR IL=2 THIS PART IS# J1(XR)*J1(X)
C                                  COMPUTED RESULTS ARE DELIVERED AS FPIGR,
C                                  EXP1 AND EXP2 IN COMMON/IGRAN/
C                                  THE SUBROUTINE CALLS IN FUNCTION BESS.
C-----------------------------------------------------------------------
      REAL LOAD,NU,ACCUR(3),K5
      COMMON/ASDT/LAYER,NLAYS,M,R,Z,NU(10),ACCUR,LOAD,HOSTRS,NZEROS,H(9)
     1,K5(10),E(10),AL(9),THICK(9),RADIUS(10)
      COMMON/IGRAN/T0,V0,S0,U0,T1,V1,S1,U1,TQ1,SQ1,FPIGR,EXP1,EXP2
      IF(LAYER.NE.1) GO TO 20
      T0 = T0-2.0*NU(1)
      V0 = V0-1.0
      T1 = T1-1.0+2.0*NU(1)
      V1 = V1+1.0
      TQ1 = TQ1-1.0
   20 IF(R.LT.ACCUR(1)) GO TO 40
      IF(IL.EQ.2) GO TO 30
      FPIGR = BESS(0,X*R)*BESS(1,X)/X
      GO TO 60
   30 FPIGR = BESS(1,X*R)*BESS(1,X)/(X*R)
      GO TO 60
   40 IF(IL.EQ.2) GO TO 50
      FPIGR = BESS(1,X)/X
      GO TO 60
   50 FPIGR = 0.5*BESS(1,X)
   60 IF(NLAYS.EQ.LAYER) GO TO 70
      IF(ABS(X*(2.0*H(LAYER)-Z)).GT.70.0)GO TO 70
      EXP1 = EXP(-X*(2.0*H(LAYER)-Z))
      IF((X*Z).GT.70.0)GO TO 90
      EXP2 = EXP(-X*Z)
      GO TO 100
 70   IF((X*Z).GT.70.0)GO TO 80
      EXP1 = 0.0
      EXP2 = EXP(-X*Z)
      GO TO 100
   80 EXP1 = 0.0
   90 EXP2 = 0.0
  100 RETURN
      END
      FUNCTION BESS(N,X)
C-----------------------------------------------------------------------
C                                  THE BESSEL FUNCTIONS JO(X) AND J1(X) ARE
C                                  EVALUATED FROM THEIR CHEBYSHEV SERIES.
C                                  (SEE CLENSHAW,MATH. TABLES-VOL.5,
C                                  CHEBYSHEV SERIES FOR MATH. FUNCTIONS
C                                  NPL-DSIR).
C                                  THIS PROGRAM SELECTS THE APPROPRIATE
C                                  CHEBYSHEV CONSTANTS ACCORDING TO WHETHER
```

```
C                                   N=0 OR N=1 AND WHETHER X IS GREATER OR
C                                   LESS THAN 8.0 AND CALLS IN FUNCTION CHEB
C                                   TO SUM THE SERIES.
C-----------------------------------------------------------------------
      DOUBLE PRECISION B(12,2),BP(5,2),BQ(5,2),Z
      DATA B /-.3D-8,.76D-7,-.1762D-5,.32460D-4,-.460626D-3,.4819180D-2,
     1-.34893769D-1,.158067102D+0,-.370094994D+0,.265178613D+0,
     2-.8723442D-2,.315455943D+0,-.1D-8,.29D-7,-.762D-6,.15887D-4,
     3-.260444D-3,.3240270D-2,-.29175525D-1,.177709117D+0,-.661443934D+0
     4,1.28799410D+0,-1.19180116D+0,1.29671754D+0/
      DATA BP/.2D-8,-.52D-7,.3075D-5,-.536522D-3,1.99892070D+0,
     1-.2D-8,.62D-7,-.3987D-5,.898990D-3,2.00180608D+0/
      DATA BQ/-.1D-8,.18D-7,-.741D-6,.68385D-4,-.31111709D-1,
     1.1D-8,-.21D-7,.914D-6,-.96277D-4,.93555574D-1/
      M = N+1
      IF(X-8.0) 1,1,2
    1 Z = X*X*0.0625-2.0
      BESS = CHEB(B(1,M),12,Z)
      IF(N.EQ.1) BESS = 0.125*X*BESS
      RETURN
    2 Z = 256.0/(X*X)-2.0
      XI = X-0.78539816
      IF(N.EQ.1) XI = XI-1.5707963
      BESS = (0.79788456/SQRT(X))*(CHEB(BP(1,M),5,Z)*COS(XI)-8.0*
     1         CHEB(BQ(1,M),5,Z)*SIN(XI)/X)
      RETURN
      END
      FUNCTION CHEB(A,N,Z)
C-----------------------------------------------------------------------
C                                   THIS SUBPROGRAM EVALUATES THE CHEBYSHEV
C                                   SERIES USING THE RECURRENCE RELATION
C                                   TECHNIQUE (SEE CLENSHAW NPL MATH. TABLES
C                                   VOLUME 5 PAGE 9).
C-----------------------------------------------------------------------
      DOUBLE PRECISION A(1),B(14),Z
      B(1)=0.0D+0
      B(2)=0.0D+0
      DO 1 I=1,N
         B(I+2)=Z*B(I+1)-B(I)+A(I)
    1 CONTINUE
      CHEB = 0.5D0*(B(N+2)-B(N))
      RETURN
      END
      REAL FUNCTION IGRAND (X,LABEL)
C-----------------------------------------------------------------------
C                                   THIS SUBROUTINE COMPUTES THE INTEGRANDS
C                                   FOR THE INTEGRALS COMPUTED IN INGRAL.
C                                   USE IS MADE OF THE RESULTS OF FPIGRA AND
C                                   MATRIX STORED IN COMMON/IGRAN/.
C-----------------------------------------------------------------------
      REAL LOAD,NU,ACCUR(3),K5
      COMMON/ASDT/LAYER,NLAYS,M,R,Z,NU(10),ACCUR,LOAD,HOSTRS,NZEROS,H(9)
     1,K5(10),E(10),AL(9),THICK(9),RADIUS(10)
```

```
      COMMON/IGRAN/T0,V0,S0,U0,T1,V1,S1,U1,TQ1,SQ1,FPIGR,EXP1,EXP2
      GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170)
     1,LABEL
   10 IGRAND =FPIGR*X*((U0*(K5(LAYER)-X*Z)-S0)*EXP1+(T0+V0*(K5(LAYER)+X*
     1Z))*EXP2)
      RETURN
   20 IGRAND =FPIGR*X*(U0*EXP1+V0*EXP2)
      RETURN
   30 IGRAND =FPIGR*((U0*(2.0*K5(LAYER)-X*Z)-S0)*EXP1-(T0+V0*(2.0*K5(LAY
     1ER)+X*Z))*EXP2)
      RETURN
   40 IGRAND =FPIGR*((S0+U0*(1.0+X*Z))*EXP1+(V0*(1.0-X*Z)-T0)*EXP2)
      RETURN
   50 IGRAND =FPIGR*X*R*((S0+U0*(2.*NU(LAYER)+X*Z))*EXP1+(T0+V0*(X*Z-2.*
     1NU(LAYER)))*EXP2)
      RETURN
   60 IGRAND =FPIGR*X*((S1+U1*(2.0*NU(LAYER)+X*Z))*EXP1+(T1+V1*(X*Z-2.0*
     1NU(LAYER)))*EXP2)
      RETURN
   70 IGRAND =FPIGR*((S1+U1*(1.0+X*Z))*EXP1+(V1*(1.0-X*Z)-T1)*EXP2)
      RETURN
   80 IGRAND =FPIGR*X*R*((S1+U1*(1.0+X*Z))*EXP1+(V1*(1.0-X*Z)-T1)*EXP2)
      RETURN
   90 IGRAND =FPIGR*X*R*(U1*EXP1+V1*EXP2)
      RETURN
  100 IGRAND =FPIGR*((S1+U1*(2.*NU(LAYER)+X*Z))*EXP1+(T1+V1*(X*Z-2.*NU(L
     1AYER)))*EXP2)
      RETURN
  110 IGRAND =FPIGR*(U1*EXP1-V1*EXP2)
      RETURN
  120 IGRAND =FPIGR*((S1+U1*(1.+X*Z))*EXP1+(V1*(1.-X*Z)-T1)*EXP2)/X
      RETURN
  130 IGRAND =FPIGR*X*(SQ1*EXP1-TQ1*EXP2)
      RETURN
  140 IGRAND =FPIGR*(SQ1*EXP1+TQ1*EXP2)
      RETURN
  150 IGRAND =FPIGR*X*R*(SQ1*EXP1+TQ1*EXP2)
      RETURN
  160 IGRAND =FPIGR*(SQ1*EXP1-TQ1*EXP2)
      RETURN
  170 IGRAND =FPIGR*(SQ1*EXP1+TQ1*EXP2)/X
      RETURN
      END
      SUBROUTINE CALC(INT,V,R,MU,RADI,FT,LOAD,HOSTRS,PSI0,Z)
C-----------------------------------------------------------------------
C                                COMPONENTS OF THE STRESSES,STRAINS AND
C                                THIS SUBROUTINE COMPUTES THE CYLINDRICAL
C                                DISPLACEMENTS FROM THE 17 INTEGRALS STO-
C                                RED IN INT. THESE CALCULATED COMPONENTS
C                                ARE STORED IN V AND OUTPUTTED.
C-----------------------------------------------------------------------
      REAL INT(17),V(15),MU,LOAD,C(6)
      INTEGER FM(19),FMT(5),T(12)
```

```
      LOGICAL STRESS,EPS,RLOW
      COMMON/STRDTA/STRESS(27),EPS(17),RLOW,ST,CT,L,ACC
      COMMON/TAPE/NOUT
      DATA FM(1),FMT,FM(19),T/
     +'(1X ',',',E12','.4,1','0X  ',',12X',' ,1','X)  ',
     +'DISP','LACE','MENT','S   ',
     +'   S','TRES','SES ',','    ',
     +'   S','TRAI','NS  ',','    '/
      DO 10 I=1,15
   10 V(I)=0.0
      IF((STRESS( 4).OR.STRESS( 5).OR.STRESS( 7).OR.STRESS(10).OR.
     +    STRESS(11)).AND.(.NOT.RLOW)) FCT=(2.0*INT(12)-INT( 7)-2.0*INT(
     + 14)+4.0*INT(17))/R
      IF(.NOT.STRESS( 1))  GO TO 20
      V( 1)=FT*RADI*CT*(2.0*INT(17)+INT(12)-INT( 7))
      IF(RLOW)             GO TO 20
      V( 1)=V( 1)-FT*R*RADI*INT( 4)
   20 IF(STRESS( 2))  V( 2)=FT*RADI*ST*(2.0*(INT(17)-INT(14))+INT(12))
      IF(.NOT.STRESS( 3))  GO TO 30
      V( 3)=-FT*RADI*INT( 3)
      IF(RLOW)             GO TO 30
      V( 3)=V( 3)+FT*R*RADI*CT*((2.0-2.0*MU)*INT(11)-INT(10))
   30 IF(.NOT.STRESS( 4))  GO TO 40
      V( 4)=CT*(INT( 8)+2.0*MU*INT( 9))+INT( 1)+INT( 4)-2.0*INT( 2)
      IF(RLOW)             GO TO 40
      V( 4)=V( 4)-CT*FCT
   40 IF(.NOT.STRESS( 5))  GO TO 50
      V( 5)=CT*2.0*MU*INT( 9)-2.0*MU*INT( 2)-INT( 4)
      IF(RLOW)             GO TO 50
      V( 5)=V( 5)+CT*FCT
   50 IF(.NOT.STRESS( 7))  GO TO 60
      V( 7)=ST*INT(15)
      IF(RLOW)             GO TO 60
      V( 7)=V( 7)-ST*FCT
   60 IF(.NOT.STRESS(10))  GO TO 70
      V(10)=FT*(CT*INT( 8)+INT( 1)+INT( 4)-(2.0-2.0*MU)*INT( 2))
      IF(RLOW)             GO TO 70
      V(10)=V(10)-FT*CT*FCT
   70 IF(.NOT.STRESS(11))  GO TO 80
      V(11)=-FT*INT( 4)
      IF(RLOW)             GO TO 80
      V(11)=V(11)+FT*CT*FCT
   80 IF(STRESS(12))  V(12)=FT*(CT*((2.0-4.0*MU)*INT( 9)-INT( 8))+
     +                          2.0*MU*INT( 2)-INT( 1))
      IF(Z.LT.ACC)  GO TO 90
      IF(STRESS( 6))  V( 6)=CT*((2.0-2.0*MU)*INT( 9)-INT( 8))-INT( 1)
      IF(STRESS( 8))  V( 8)=CT*(INT(16)+INT(10)-INT( 6))-INT( 5)
      IF(STRESS( 9))  V( 9)=ST*(INT(16)-INT(13)+INT(10))
      GO TO 110
   90 IF(ABS(R-1.0).LT.ACC)  GO TO 100
      IF(R.GT.1.0)         GO TO 110
      IF(STRESS( 6))  V( 6)=-LOAD
      IF(STRESS( 8))  V( 8)=-HOSTRS*COS(PSIO)
```

```
      IF(STRESS( 9))  V( 9)=HOSTRS*SIN(PSIO)
      GO TO 110
  100 IF(STRESS( 6))  V( 6)=-0.5*LOAD
      IF(STRESS( 8))  V( 8)=-0.5*HOSTRS*COS(PSIO)
      IF(STRESS( 9))  V( 9)= 0.5*HOSTRS*SIN(PSIO)
  110 IF(STRESS(13))  V(13)=FT*V( 7)
      IF(STRESS(14))  V(14)=FT*V( 8)
      IF(STRESS(15))  V(15)=FT*V( 9)
      DO 120 I=2,18
  120 FM(I)=T(8)
      DO 130 I=4,16,3
  130 FM(I)=FMT(3)
      K=0
      J=0
      DO 210 I=1,15
         J=J+1
         IF(I-4)  190,160,140
  140    IF(I-10)  190,150,190
  150    IF(K.EQ.0)   GO TO 180
         WRITE(NOUT,9010) (T(J),J=5,8)
         WRITE(NOUT,FM) (C(J),J=1,K)
         GO TO 170
  160    IF(K.EQ.0)   GO TO 180
         WRITE(NOUT,9000) (T(J),J=1,4)
      WRITE(NOUT,9013) (C(J),J=1,K)
  170    K=0
  180    J=1
  190    M=3*J
         IF(.NOT.STRESS(I))  GO TO 200
         K=K+1
         C(K)=V(I)
         FM(M-1)=FMT(1)
         FM(M  )=FMT(2)
         GO TO 210
  200    FM(M-1)=FMT(4)
         FM(M  )=FMT(5)
  210 CONTINUE
      IF(K.EQ.0)  RETURN
      WRITE(NOUT,9010) (T(J),J=9,12)
      WRITE(NOUT,FM) (C(J),J=1,K)
      RETURN
 9000 FORMAT(1X,4A4/5X,'RADIAL',12X,'TANGENTIAL',14X,'VERTICAL')
 9010 FORMAT(1X,4A4/5X,'RADIAL',12X,'TANGENTIAL',14X,'VERTICAL',12X,'RAD
     +./TANG.',11X,'RAD./VERT.',12X,'TANG./VERT.')
 9013 FORMAT(1X,3(E12.4,10X))
      END
      SUBROUTINE OUTPUT(EPS,C,K,L)
C-----------------------------------------------------------------------
C                                   THIS SUBROUTINE OUTPUTS BY MEANS OF THREE
C                                   SUBSEQUENT CALLS FROM THE MAIN PROGRAM
C                                   THE TOTAL STRESSES,STRAINS AND SISPLACE-
C                                   MENTS.
C-----------------------------------------------------------------------
```

```
      INTEGER FM(16),FMT(8)
      LOGICAL EPS(6)
      DIMENSION C(6),TKST(6,4)
      COMMON/TAPE/NOUT
      DATA TKST/
     1' T O',' T A',' L  ','S T ','R E ','S S ',
     2' T O',' T A',' L  ','S T ','R A ','I N ',
     3' T O',' T A',' L  ','D I ','S P ','L A ',
     4'C E ','M E ','N T ','    ','    ','    '/
      DATA FMT,FM(16)/
     1'(6A4','.12X','    ',',E12','.3  ','(12A','4,48','X   ',')   '/
      IF(L.NE.3) GO TO 10
      FM(1)=FMT(6)
      FM(2)=FMT(7)
      FM(3)=FMT(8)
      GO TO 20
   10 FM(1)=FMT(1)
      FM(2)=FMT(3)
      FM(3)=FMT(3)
   20 N=0
      M=2*K+2
      DO 40 I=4,M,2
         J=I/2-1
         IF(.NOT.EPS(J)) GO TO 30
         FM(I)=FMT(4)
         FM(I+1)=FMT(5)
         N=N+1
         C(N)=C(J)
         GO TO 40
   30    FM(I)=FMT(2)
         FM(I+1)=FMT(3)
   40 CONTINUE
      IF(L.EQ.3) GO TO 60
      IF(N.EQ.0) GO TO 50
      WRITE(NOUT,FM) (TKST(I,L),I=1,6),(C(I),I=1,N)
      RETURN
   50 WRITE(NOUT,FM) (TKST(I,L),I=1,6)
      RETURN
   60 DO 70 I=10,15
   70 FM(I)=FMT(3)
      IF(N.EQ.0) GO TO 80
      WRITE(NOUT,FM) (TKST(I,3),I=1,6),(TKST(I,4),I=1,6),(C(I),I=1,N)
      RETURN
   80 WRITE(NOUT,FM) (TKST(I,3),I=1,6),(TKST(I,4),I=1,6)
      RETURN
      END
      SUBROUTINE JACOBI (H,U,ND,N,IVEC,W,IQ)
C-----------------------------------------------------------------------
C                                SUBROUTINE JACOBI TO COMPUTE EIGENVALUES
C                                AND EIGENVECTORS OF A SYMMETRIC MATRIX.
C                                H IS THE GIVEN MATRIX,THE DIAGONAL OF
C                                WHICH CONTAINS AFTER THE ITERATION THE
C                                EIGENVALUES OF H.
```

```
C                               U IS THE MATRIX,THE COLUMNS OF WHICH ARE
C                               THE EIGENVECTORS OF H.
C                               N AND ND ARE THE DIMENSIONS OF THE ACTUAL
C                               MATRIX AND THE ONE USED IN THE DIMENSION-
C                               STATEMENT OF THE CALLINGPROGRAM
C                               RESPECTIVELY.
C                               IVEC=0 IF NO EIGENVECTORS ARE REQUIRED,
C                               IVEC=1 IF THE EIGENVECTORS SHOULD BE
C                               CALCULATED.
C                               THE ACCURACY OF THE EIGENVALUES IS ABOUT
C                               1.0E-6,THE ACCURACY OF AN EIGENVECTOR IS
C                               ABOUT 1.0E-6/D,WHERE D IS THE MINIMUM-
C                               DISTANCE OF THE CORRESPONDING EIGENVALUE
C                               FROM THE OTHER EIGENVALUES.
C                               W AND IQ ARE WORKINGSPACES,WHICH SHOULD BE
C                               DIMENSIONED IN THE CALLING PROGRAM.
C-----------------------------------------------------------------------
      REAL H(ND,ND),U(ND,ND),W(ND)
      INTEGER IQ(ND)
      DOUBLE PRECISION TA,SI,CO,Z,Y,HTE,UTE
      AN   =N
      NMI1=N-1
      IF(IVEC-1) 60,10,60
   10 DO  50 I=1,N
         DO  40 J=1,N
            IF(I-J) 30,20,30
   20       U(I,J)=1.0
            GO TO 40
   30       U(I,J)=0.0
   40    CONTINUE
   50 CONTINUE
   60 DO  90 I=1,NMI1
         W(I)=0.0
         IPL1=I+1
         DO  80 J=IPL1,N
            IF(W(I)-ABS(H(I,J))) 70,70,80
   70       W(I)=ABS(H(I,J))
            IQ(I)=J
   80    CONTINUE
   90 CONTINUE
  100 DO 120 I=1,NMI1
         IF(I.EQ.1) GO TO 110
         IF(XMAX.GE.W(I)) GO TO 120
  110    XMAX=W(I)
         IPIV=I
         JPIV=IQ(I)
  120 CONTINUE
      IF(XMAX-1.E-12/AN) 170,170,130
  130 Z  =H(IPIV,IPIV)-H(JPIV,JPIV)
      Y = 2.0D0*DBLE(H(IPIV,JPIV))
      TA =Y/(DABS(Z)+DSQRT(Z*Z+Y*Y))
      IF(Z.LT.0.0D0)  TA=-TA
      CO =1.D0/DSQRT(1.D0+TA*TA)
```

```
      SI =TA*CO
      HII=H(IPIV,IPIV)
      HJJ=H(JPIV,JPIV)
      HIJ=H(IPIV,JPIV)
      DO 140 K=1,N
         HTE=H(K,IPIV)
         H(K,IPIV)=DBLE(H(K,IPIV))*CO+DBLE(H(K,JPIV))*SI
         H(K,JPIV)=DBLE(H(K,JPIV))*CO-HTE*SI
         H(IPIV,K)=H(K,IPIV)
         H(JPIV,K)=H(K,JPIV)
  140 CONTINUE
      H(IPIV,JPIV)=0.0
      H(JPIV,IPIV)=0.0
      AA=DBLE(HIJ)*TA
      H(IPIV,IPIV)=HII+AA
      H(JPIV,JPIV)=HJJ-AA
      IF(IVEC)  60,60,150
  150 DO 160 K=1,N
         UTE=U(K,IPIV)
         U(K,IPIV)=DBLE(U(K,IPIV))*CO+DBLE(U(K,JPIV))*SI
         U(K,JPIV)=DBLE(U(K,JPIV))*CO-UTE*SI
  160 CONTINUE
      GO TO  60
  170 RETURN
      END
      SUBROUTINE ESORT (H,U,ND,N,IVEC,W,IQ)
C------------------------------------------------------------------------
C                                   SUBROUTINE ESORT.
C                                   THIS ROUTINE SORTS EIGENVALUES (AND EIGEN
C                                   VECTORS) OBTAINED FROM SUBROUTINE JACOBI.
C                                   H    = ORIGINAL MATRIX(ND,ND).
C                                   U    = EIGENVECTORMATRIX(ND,ND).
C                                   ND   = MAX. DIMENSION OF MATRICES.
C                                   N    = ACTUAL DIMENSION OF MATRICES.
C                                   IVEC=1 WITH EIGENVECTORS,
C                                       =0   NO EIGENVECTORS.
C                                   W    = WORKINGSPACE(ND).
C                                   IQ   = WORKINGSPACE(ND).
C------------------------------------------------------------------------
      REAL H(ND,ND),U(ND,ND),W(ND),DUMMY
      INTEGER N,IQ(ND),FDUMMY,I,J,K,IVEC
      LOGICAL LOGIC
C
      DO 10 I=1,N
         W(I)=H(I,I)
   10    IQ(I)=I
      J=N
   20 LOGIC=.FALSE.
      K=J
      DO 30 I=2,K
         IF(W(I-1).GE.W(I)) GO TO 30
         LOGIC=.TRUE.
         DUMMY=W(I-1)
```

```
          W(I-1)=W(I)
          W(I)=DUMMY
          FDUMMY=IQ(I-1)
          IQ(I-1)=IQ(I)
          IQ(I)=FDUMMY
          J=I-1
   30 CONTINUE
      IF (LOGIC) GO TO 20
      IF (IVEC.EQ.0) GO TO 60
      DO 40 I=1,N
         K=IQ(I)
         DO 40 J=1,N
   40       H(J,I)=U(J,K)
      DO 50 I=1,N
         DO 50 J=1,N
            U(I,J)=H(I,J)
   50       H(I,J)=0.0
   60 DO 70 I=1,N
   70    H(I,I)=W(I)
      RETURN
      END
      SUBROUTINE RNN(AA,BB,RN,ALOAD,ALIN,CAREA)
C     FLEXIBLE PAVEMENT ROUTINE
C     COMPUTES THE LIMITING VERTICAL STRAIN AS A FUNCTION OF LOAD
C     REPETITION, INDEPENDENT OF THE SUBGRADE YOUNG'S MODULUS.
      COMMON VSTR,STRL,ITER,STRL2
      COMMON/RADIAL/STSL,DSM,FS,SWL
      WRITE(6,10)
   10 FORMAT(//'PROGRAM USING STRAIN REPETITION NUMBER ..........'//)
      CALL FFRD(5,AA,1)
      CALL FFRD(5,BB,2)
      CALL FFRD(5,RN,2)
      CALL FFRD(5,ALOAD,2)
      CALL FFRD(5,ALIN,2)
      CALL FFRD(5,CAREA,2)
      CALL FFRD(5,DSW,2)
      CALL FFRD(5,SWL,2)
      YY=ALOG10(RN)
      Q=AA*YY+BB
      STRL=10.**Q
      RETURN
      END
      SUBROUTINE RPAL(ALOAD,ALIN,CAREA)
C RIGID PAVEMENT ALLOWABLE LOAD ROUTINE.
C     COMPUTES THE LIMITING RADIAL STRAIN.
      COMMON/RADIAL/STSL,DSM,FS,SWL
      DATA A,B/0.58901,0.35486/
      CALL FFRD(5,DSM,1)
      CALL FFRD(5,FAC,2)
      CALL FFRD(5,YRN,2)
      CALL FFRD(5,FS ,2)
      CALL FFRD(5,ALOAD,2)
      CALL FFRD(5,ALIN,2)
```

```
      CALL FFRD(5,CAREA,2)
      CALL FFRD(5,SWL,2)
      RN=20.*YRN                                                        00036710
      COV=RN/FAC                                                        00036720
      STSL=FS/(A+B*ALOG10(COV))                                         00036730
      WRITE(6,2)A,B,YRN,FAC,COV                                         00036740
    2 FORMAT(////60(1H*)/,20X,'A   = ',F10.6//,                         00036760
     1                    20X,'B   = ',F10.6//,                         00036770
     2                    20X,'YRN = ',F10.0//,                         00036780
     3                    20X,'FAC = ',F10.4//,                         00036790
     4                    20X,'COV = ',F10.0,/60(1H*)////)              00036800
      RETURN                                                            00036810
      END                                                               00036820
      SUBROUTINE FLEX(ES,EA,ALOAD,ALIN,CAREA,XS,AS,BS,YRN,PRATIO)       +0036830
C     FLEXIBLE PAVEMENT ROUTINE                                         00036840
      DIMENSION AO(8),BO(8)                                             00036850
      COMMON VSTR,STRL,ITER,STRL2                                       00036860
      COMMON /RADIAL/STSL,DSM,FS,SWL                                    00036870
      DATA AO/2.32E-03,-9.6347304E-05,2.9507649E-05,-4.2586473E-06,     00036880
     13.0210971E-07,-1.1453280E-08,2.2311010E-10,-1.7523733E-12/        00036890
      DATA BO/1.54E-01,-1.7032892E-02,2.8018403E-03,-3.5283369E-04,     00036900
     12.5877567E-05,-1.0485427E-06,2.1756275E-08,-1.7990976E-10/        00036910
      CALL FFRD(5,ES,1)
      CALL FFRD(5,EA,2)
      CALL FFRD(5,YRN,2)
      CALL FFRD(5,ALOAD,2)
      CALL FFRD(5,ALIN,2)
      CALL FFRD(5,CAREA,2)
      CALL FFRD(5,DSM,2)
      CALL FFRD(5,SWL,2)
      CALL FFRD(5,PRATIO,2)
      WRITE(6,11)YRN                                                    00036940
   11 FORMAT(//,'PROGRAM USING ',F10.0,3X,'REPETITIONS.........',//)
      XS=ES/1000.                                                       00036960
      AS=AO(1)+AO(2)*XS+AO(3)*XS**2+AO(4)*XS**3+AO(5)*XS**4+            00036970
     1AO(6)*XS**5+AO(7)*XS**6+AO(8)*XS**7                               00036980
      BS=BO(1)+BO(2)*XS+BO(3)*XS**2+BO(4)*XS**3+BO(5)*XS**4+BO(6)       00036990
     1*XS**5+BO(7)*XS**6+BO(8)*XS**7                                    00037000
       BSNEG=-BS                                                        00037010
       STRL=AS*YRN**BSNEG                                               00037020
      ACREPS=ALOG10(YRN*20./PRATIO)                                     +0037030
      EA1=EA/14.22                                                      00037040
      EALOG=ALOG10(EA1)                                                 00037050
      STRL2=(ACREPS+2.665*EALOG+0.392)/5.                               00037060
      STRL2=-STRL2                                                      00037070
      STRL2=10.**STRL2                                                  00037080
      RETURN                                                            00037090
      END                                                               00037100
      SUBROUTINE POST(XTEMP,YTEMP,NPOS2,LAY,AXX,AYY,DEP,ETAA)           00037110
C     ROUTINE THAT GENERATES THE COMPUTATIONAL POSITIONS ONCE THE       00037120
C     FINAL LOAD OR THICKNESS IS FOUND.                                 00037130
      REAL NU,ACCUR(3),LOAD,K5                                          00037140
      COMMON/ASDT/LAYER,NLAYS,M,R,Z,NU(10),ACCUR,LOAD,HOSTRS,NZEROS,H(9)00037150
```

```
     +,K5(10),E(10),AL(9),THICK(9),RADIUS(10)
      DIMENSION XTEMP(10),YTEMP(10),LAY(100),AXX(100),AYY(100),
     1          DEP(100),ETAA(100)
      NLOC=NPOS2
      NPOS2=NLAYS*2-1
      M1=2
      M2=NPOS2
      KPOS=0
      K=0
      WRITE(6,13)THICK
   13 FORMAT(//'THICK ARRAY = ',/9F10.2//)
      DO 48 I=1,NLOC
           LAY(K+1)=1
           AXX(K+1)=XTEMP(I)
           AYY(K+1)=YTEMP(I)
           DEP(K+1)=0.
           ETAA(K+1)=0.
           KK=1
           TSUM=0.
                DO 49 J=M1,M2,2
                     LAY(J)=(J-KPOS)/2
                     LAY(J+1)=(J-KPOS)/2+1
                     AXX(J)=XTEMP(I)
                     AXX(J+1)=XTEMP(I)
                     AYY(J)=YTEMP(I)
                     AYY(J+1)=YTEMP(I)
      WRITE(6,12)KK,THICK(KK)
   12 FORMAT('KK = ',I5,5X,'THICK = ',F8.2//)
                     DEP(J)=THICK(KK)+TSUM
                     DEP(J+1)=THICK(KK)+TSUM
                     ETAA(J)=0.
                     ETAA(J+1)=0.
                     KK=KK+1
                     TSUM=DEP(J)
   49           CONTINUE
           IF(I.EQ.NLOC)GO TO 48
           K=NPOS2+K
           M1=M2+2
           M2=NPOS2+M2
           KPOS=KPOS+NPOS2
   48 CONTINUE
      NPOS2=M2
      WRITE(6,10)
      WRITE(6,11)(LAY(I),AXX(I),AYY(I),DEP(I),ETAA(I),I=1,M2)
   10 FORMAT('SUBROUTINE POST--------')
   11 FORMAT((I10,4F15.2/))
      RETURN
      END
      SUBROUTINE NFRD(IFC,NV,ID)
      COMMON/JRJ/ICRD(80),IFMT(10),IVAL(10),NL
      DATA IB/1H /,IC/1H,/
      IF(ID.NE.1) GO TO 10
      READ(IFC,2) (ICRD(I),I=1,80)
```

```
    2 FORMAT(80A1)
      NL=1
   10 IF(NL.GE.80) NV=0
      IF(NL.GE.80) RETURN
C
C     FIND FIRST NONE BLANK CHARACTER
C
      DO 3 J=NL,80
      IF(ICRD(J).NE.IB) GO TO 5
      I=J
    3 CONTINUE
      NL=I+1
      GO TO 10
    5 NL=J
      IF(ICRD(NL).NE.IC) GO TO 4
      NV=0
      NL=NL+1
      RETURN
    4 DO 6 J=NL,80
      IF(ICRD(J).EQ.IB) GO TO 7
      IF(ICRD(J).EQ.IC) GO TO 7
    6 CONTINUE
       J=81
    7 K=NL
      M=J-1
      ENCODE(40,2,IVAL(1)) (ICRD(I),I=K,M)
      NL=J+1
      L=M-K+1
      IF(L.LE.9) ENCODE(40,8,IFMT(1)) L
      IF(L.GE.10) ENCODE(40,9,IFMT(1)) L
    8 FORMAT('(I',I1,')')
    9 FORMAT('(I',I2,')')
      DECODE(L,IFMT,IVAL(1)) NV
      RETURN
      END
      SUBROUTINE FFRD(IFC,F,ID)
      COMMON/JRJ/ICRD(80),IFMT(10),IVAL(10),NL
      DATA IB/1H /,IC/1H,/
      IF(ID.NE.1) GO TO 10
      READ(IFC,2) (ICRD(I),I=1,80)
    2 FORMAT(80A1)
      NL=1
   10 IF(NL.GE.80) F=0.0
      IF(NL.GE.80) RETURN
C
C     FIND FIRST NONE BLANK CHARACTER
C
      DO 3 J=NL,80
      IF(ICRD(J).NE.IB) GO TO 5
      I=J
    3 CONTINUE
      NL=I+1
      GO TO 10
```

```
      5 NL=J
        IF(ICRD(NL).NE.IC) GO TO 4
        F=0.0
        NL=NL+1
        RETURN
      4 DO 6 J=NL,80
        IF(ICRD(J).EQ.IB) GO TO 7
        IF(ICRD(J).EQ.IC) GO TO 7
      6 CONTINUE
         J=81
      7 K=NL
        M=J-1
        ENCODE(40,2,IVAL(1)) (ICRD(I),I=K,M)
        NL=J+1
        L=M-K+1
        IF(L.LE.9) ENCODE(40,8,IFMT(1)) L
        IF(L.GE.10) ENCODE(40,9,IFMT(1)) L
      8 FORMAT('(F',I1,'.0)')
      9 FORMAT('(F',I2,'.0)')
        DECODE(L,IFMT,IVAL(1)) F
        RETURN
        END
        BLOCK DATA                                                      00037640
C-----------------------------------------------------------------------00037650
C                                 IN THE BLOCK DATA ARE STORED SUBSEQUNTILY*00037660
C                                 -THE ABSCISSAE FOR THE LEGENDRE-GAUSS     00037670
C                                  QUADRATURE,STARTING IN A WITH THE 2-ND   00037680
C                                  ORDER AND ENDING IN N WITH THE 15-TH     00037690
C                                  ORDER.                                   00037700
C                                 -THE ABSCISSAE FOR THE JACOBI-GAUSS       00037710
C                                  QUADRATURE OF THE 8-TH ORDER IN O.       00037720
C                                 -THE WEIGHTS FOR THE LEGENDRE-GAUSS       00037730
C                                  QUADRATURE,STARTING IN P WITH THE 2-ND   00037740
C                                  ORDER AND ENDING IN CC WITH THE 15-TH    00037750
C                                  ORDER.                                   00037760
C                                 -THE WEIGHTS FOR THE JACOBI-GAUSS QUADRA- 00037770
C                                  TURE OF THE 8-TH ORDER IN DD.            00037780
C                                 -THE FIRST 149 ZEROS OF J0 IN EE AND FF   00037790
C                                 -THE FIRST 149 ZEROS OF J1 IN GG AND HH   00037800
C-----------------------------------------------------------------------00037810
        REAL I,J,K,L,M,N                                                00037820
        DIMENSION A(2),B(3),C(4),D(5),E(6),F(7),G(8),H(9),I(10),J(11),K(1200037830
       1),L(13),M(14),N(15),O( 8),P(2),Q(3),R(4),S(5),T(6),U(7),V(8),W(9),00037840
       2X(10),Y(11),Z(12),AA(13),BB(14),CC(15),DD( 8),EE(119),FF(30),GG(1100037850
       39),HH(30)                                                       00037860
        COMMON/GAUSS/AGAUSS(16,16),HGAUSS(16,16)                        00037870
        COMMON/GEDATA/BZEROS(149,2),ZEROS(298)                          00037880
        EQUIVALENCE            (AGAUSS(1, 2),  A(1)),(AGAUSS(1, 3),  B(1)),  00037890
       1(AGAUSS(1, 4),  C(1)),(AGAUSS(1, 5),  D(1)),(AGAUSS(1, 6),  E(1)),  00037900
       2(AGAUSS(1, 7),  F(1)),(AGAUSS(1, 8),  G(1)),(AGAUSS(1, 9),  H(1)),  00037910
       3(AGAUSS(1,10),  I(1)),(AGAUSS(1,11),  J(1)),(AGAUSS(1,12),  K(1)),  00037920
       4(AGAUSS(1,13),  L(1)),(AGAUSS(1,14),  M(1)),(AGAUSS(1,15),  N(1)),  00037930
       5(AGAUSS(1,16),  O(1)),(HGAUSS(1, 2),  P(1)),(HGAUSS(1, 3),  Q(1)),  00037940
```

```
     6(HGAUSS(1, 4), R(1)),(HGAUSS(1, 5), S(1)),(HGAUSS(1, 6), T(1)),      00037950
     7(HGAUSS(1, 7), U(1)),(HGAUSS(1, 8), V(1)),(HGAUSS(1, 9), W(1)),      00037960
     8(HGAUSS(1,10), X(1)),(HGAUSS(1,11), Y(1)),(HGAUSS(1,12), Z(1)),      00037970
     9(HGAUSS(1,13),AA(1)),(HGAUSS(1,14),BB(1)),(HGAUSS(1,15),CC(1)),      00037980
     T(HGAUSS(1,16),DD(1)),(BZEROS(1, 1),EE(1)),(BZEROS(120,1),FF(1)),     00037990
     1(BZEROS(1, 2),GG(1)),(BZEROS(120,2),HH(1))                           00038000
      DATA A,B,C,D,E,F,G,H,I,J,K,L,M                                       00038010
     N/-.4472136, 0.4472136,-0.6546537, 0.0000000, 0.6546537,-0.7650554,00038020
     1-0.2852315, 0.2852315, 0.7650553,-0.8302239,-0.4688488, 0.0000000,00038030
     2 0.4688488, 0.8302239,-0.8717402,-0.5917003,-0.2092993, 0.2092991,00038040
     3 0.5917001, 0.8717400,-0.8997580,-0.6771863,-0.3631175, 0.0000000,00038050
     4 0.3631175, 0.6771863, 0.8997580,-0.9195339,-0.7387739,-0.4779250,00038060
     5-0.1652790, 0.1652789, 0.4779249, 0.7387738, 0.9195338,-0.9340014,00038070
     6-0.7844836,-0.5652354,-0.2957582, 0.0000000, 0.2957581, 0.5652353,00038080
     7 0.7844834, 0.9340014,-0.9448975,-0.8192815,-0.6328754,-0.3995310,00038090
     8-0.1365529, 0.1365529, 0.3995309, 0.6328753, 0.8192813, 0.9448975,00038100
     9-0.9533069,-0.8463538,-0.6861843,-0.4829108,-0.2492869, 0.0000000,00038110
     T 0.2492868, 0.4829106, 0.6861842, 0.8463537, 0.9533068,-0.9599299,00038120
     1-0.8678104,-0.7288621,-0.5506417,-0.3427235,-0.1163319, 0.1163318,00038130
     2 0.3427235, 0.5506415, 0.7288620, 0.8678104, 0.9599298,-0.9652544,00038140
     3-0.8850636,-0.7635341,-0.6062477,-0.4206389,-0.2153539, 0.0000000,00038150
     4 0.2153538, 0.4206389, 0.6062477, 0.7635341, 0.8850635, 0.9652544,00038160
     5-0.9695861,-0.8991729,-0.7920153,-0.6523931,-0.4860575,-0.2998304,00038170
     6-0.1013263, 0.1013262, 0.2998304, 0.4860575, 0.6523930, 0.7920151,00038180
     7 0.8991728, 0.9695860/                                               00038190
      DATA N,O                                                             00038200
     N/-.9731405,-0.9108602,-0.8157166,-0.6910172,-0.5413883,-0.3721744,00038210
     1-0.1895120, 0.0000000, 0.1895119, 0.3721744, 0.5413882, 0.6910170,00038220
     2 0.8157164, 0.9108602, 0.9731404,-0.9602899,-0.7966665,-0.5255324,00038230
     3-0.1834346, 0.1834346, 0.5255324, 0.7966665, 0.9602899/              00038240
      DATA P,Q,R,S,T,U,V,W,X,Y,Z,AA,BB                                     00038250
     N/0.8333334, 0.8333331, 0.5444443, 0.7111111, 0.5444444, 0.3784749,00038260
     1 0.5548583, 0.5548581, 0.3784750, 0.2768261, 0.4317453, 0.4876190,00038270
     2 0.4317455, 0.2768261, 0.2107044, 0.3411230, 0.4124591, 0.4124591,00038280
     3 0.3411230, 0.2107046, 0.1654953, 0.2745391, 0.3464290, 0.3715193,00038290
     4 0.3464290, 0.2745388, 0.1654955, 0.1333061, 0.2248897, 0.2920431,00038300
     5 0.3275404, 0.3275403, 0.2920429, 0.2248897, 0.1333061, 0.1096126,00038310
     6 0.1871701, 0.2480485, 0.2868792, 0.3002176, 0.2868798, 0.2480485,00038320
     7 0.1871700, 0.1096126, 0.0916847, 0.1579750, 0.2125089, 0.2512758,00038330
     8 0.2714060, 0.2714060, 0.2512759, 0.2125089, 0.1579748, 0.0916846,00038340
     9 0.0778019, 0.1349820, 0.1836473, 0.2207679, 0.2440163, 0.2519308,00038350
     T 0.2440165, 0.2207679, 0.1836473, 0.1349820, 0.0778019, 0.0668373,00038360
     1 0.1165870, 0.1600221, 0.1948268, 0.2191266, 0.2316136, 0.2316138,00038370
     2 0.2191266, 0.1948268, 0.1600223, 0.1165869, 0.0668375, 0.0580301,00038380
     3 0.1016605, 0.1405119, 0.1727902, 0.1969877, 0.2119743, 0.2170480,00038390
     4 0.2119743, 0.1969876, 0.1727903, 0.1405120, 0.1016601, 0.0580301,00038400
     5 0.0508505, 0.0893939, 0.1242559, 0.1540275, 0.1774924, 0.1936907,00038410
     6 0.2019594, 0.2019594, 0.1936906, 0.1774925, 0.1540275, 0.1242554,00038420
     7 0.0893940, 0.0508506/                                               00038430
      DATA CC,DD                                                           00038440
     N/0.0449221, 0.0791985, 0.1105931, 0.1379879, 0.1603954, 0.1770052,00038450
     1 0.1872171, 0.1906618, 0.1872172, 0.1770049, 0.1603951, 0.1379883,00038460
     2 0.1105928, 0.0791985, 0.0449221, 0.1012285, 0.2223810, 0.3137067,00038470
```

```
     3 0.3626838, 0.3626838, 0.3137067, 0.2223810, 0.1012285/
      DATA EE    /2.404826, 5.520078, 8.653728, 11.79153, 14.93092,
     1 18.07106, 21.21164, 24.35247, 27.49348,  30.63461, 33.77582,
     2 36.91710, 40.05843, 43.19979, 46.34119,  49.48261, 52.62405,
     3 55.76551, 58.90698, 62.04847, 65.18996,  68.33147, 71.47298,
     4 74.61450, 77.75603, 80.89756, 84.03909,  87.18063, 90.32217,
     5 93.46372, 96.60527, 99.74682,102.8883,  106.0299, 109.1715,
     6112.3131, 115.4546, 118.5962, 121.7377,  124.8793, 128.0209,
     7131.1624, 134.3040, 137.4456, 140.5872,  143.7287, 146.8703,
     8150.0119, 153.1535, 156.2950, 159.4366,  162.5782, 165.7198,
     9168.8613, 172.0029, 175.1445, 178.2861,  181.4277, 184.5692,
     T187.7108, 190.8524, 193.9940, 197.1356,  200.2772, 203.4187,
     1206.5603, 209.7019, 212.8435, 215.9850,  219.1267, 222.2682,
     2225.4098, 228.5514, 231.6930, 234.8346,  237.9762, 241.1178,
     3244.2593, 247.4009, 250.5425, 253.6841,  256.8257, 259.9673,
     4263.1089, 266.2504, 269.3920, 272.5336,  275.6752, 278.8168,
     5281.9584, 285.1000, 288.2416, 291.3831,  294.5247, 297.6663,
     6300.8079, 303.9495, 307.0911, 310.2327,  313.3743, 316.5159,
     7319.6574, 322.7990, 325.9406, 329.0822,  332.2238, 335.3654,
     8338.5070, 341.6486, 344.7902, 347.9317,  351.0733, 354.2149,
     9357.3565, 360.4981, 363.6397, 366.7813,  369.9229, 373.0645/
      DATA FF / 376.2061, 379.3476, 382.4892,  385.6308, 388.7724,
     1391.9140, 395.0556, 398.1972, 401.3388,  404.4804, 407.6220,
     2410.7635, 413.9051, 417.0467, 420.1883,  423.3299, 426.4715,
     3429.6131, 432.7547, 435.8963, 439.0379,  442.1794, 445.3210,
     4448.4626, 451.6042, 454.7458, 457.8874,  461.0290, 464.1706,
     5467.3122/
      DATA GG    /3.831706, 7.015587, 10.17347, 13.32369, 16.47063,
     1 19.61586, 22.76008, 25.90367,  29.04683, 32.18968, 35.33231,
     2 38.47477, 41.61709, 44.75932,  47.90146, 51.04354, 54.18555,
     3 57.32753, 60.46946, 63.61136,  66.75323, 69.89507, 73.03690,
     4 76.17870, 79.32049, 82.46226,  85.60402, 88.74577, 91.88750,
     5 95.02923, 98.17095,101.3127,  104.454   107.5961, 110.7376,
     6113.8794, 117.0211, 120.1628,  123.3045, 126.4461, 129.5878,
     7132.7295, 135.8711, 139.0128,  142.1544, 145.2961, 148.4377,
     8151.5794, 154.7210, 157.8626,  161.0043, 164.1459, 167.2876,
     9170.4292, 173.5708, 176.7125,  179.8541, 182.9957, 186.1374,
     T189.2790, 192.4206, 195.5622,  198.7038, 201.8455, 204.9871,
     1208.1287, 211.2703, 214.4120,  217.5536, 220.6952, 223.8368,
     2226.9784, 230.1200, 233.2616,  236.4033, 239.5449, 242.6865,
     3245.8281, 248.9697, 252.1113,  255.2529, 258.3945, 261.5362,
     4264.6778, 267.8194, 270.9610,  274.1026, 277.2442, 280.3858,
     5283.5274, 286.6690, 289.8106,  292.9522, 296.0938, 299.2354,
     6302.3771, 305.5187, 308.6603,  311.8019, 314.9435, 318.0851,
     7321.2267, 324.3683, 327.5099,  330.6515, 333.7931, 336.9347,
     8340.0763, 343.2179, 346.3595,  349.5011, 352.6427, 355.7843,
     9358.9259, 362.0675, 365.2091,  368.3507, 371.4923, 374.6339/
      DATA HH / 377.7755, 380.9171,  384.0587, 387.2003, 390.3419,
     1393.4835, 396.6251, 399.7667,  402.9083, 406.0499, 409.1919,
     2412.3331, 415.4747, 418.6163,  421.7579, 424.8995, 428.0411,
     3431.1827, 434.3243, 437.4659,  440.6075, 443.7491, 446.8907,
     4450.0323, 453.1739, 456.3155,  459.4570, 462.5987, 465.7403,
     5468.8819/
      END
```

ATE

LMED

-8